PETR HOŠEK

AUTOMATION OF MEASUREMENT OF DYNAMIC RESPONSE OF VALVETRAINS

Phd Thesis

Technical university of Liberec, 2011

Phd Thesis

Automation of measurement of dynamic response of valvetrains

Automatizace měření dynamické odezvy ventilových rozvodů

Author: Petr Hošek
(petr.hosek@tul.cz)
Supervisor: Doc. Ing. Antoním Potěšil, CSc.
Study programme: P2612 Electrotechnics and informatics
Field of study: 2612V045 Technical Cybernetics
Department: Faculty of Mechatronics, Informatics and Interdisciplinary studies,
Institute of New Technologies and Applied Informatics,
Technical University of Liberec,
Studentská 2, 461 17 Liberec 1, Czech Republic

Prohlášení

Byl jsem seznámen s tím, že na mou disertační práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé disertační práce pro vnitřní potřebu TUL.

Disertační práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím disertační práce a konzultanty.

Datum: ……. Podpis: …….

Abstract

This thesis describes the development of a new generation of an apparatus for automated measurement of the dynamic response of valvetrains and presents analyses of the measured data. It focuses on, as a key component, the development of the measurement software and the algorithm for signal drop-out recognition that made this fully automated measurement possible. As a result the time of the measurement was drastically reduced thus increasing the throughput and improving the repeatability. The construction of the device and the software are versatile and applicable in several other measurement setups.

An implementation of Siemens USS protocol into LabVIEW is presented, as there was a permanent demand from the community of users to have the native LabVIEW implementation. The protocol is used as a communication interface to drive Siemens frequency inverter. By using the connected electromotor, the frequency inverter drives the crankshaft of the combustion engine. The encountered problems and their solutions are mentioned.

The next study introduces an algorithm for the recognition of signal drop-outs developed particularly for measurements of valvetrain kinematics performed by laser Doppler vibrometer. Signal drop-outs are the result of low amplitude levels of the Doppler signal due to changes in the speckle noise during tilt and rotation of valve. The recognition algorithm is needed in order to save data that are not affected by this phenomenon. Such an algorithm increases the throughput of the engine test stand and decreases the time needed for the following evaluation of the valvetrains of combustion engines. The work shows the most common encountered drop-outs and their characteristics and locations. It presents an automatic separation algorithm for the measured signal so that the drop-out recognition tests can be aimed at specific data intervals (valve opening, valve closing, etc.) with specifically set parameters of the algorithm.

Advanced techniques for real-time parallel measurement, software controlling, data acquisition, multi-instrument communication and post-processing were combined and are described. They assure the automation and certain level of intelligence and safety of the developed system.

To evaluate the developed system and to show its overall capabilities, first the standard ŠKODA 1.2 HTP engine with OHC valvetrain was measured. Second the low-friction valvetrain for the same engine was examined and compared. A direct comparison of valve acceleration calculated while presuming constant camshaft speed and the approach that benefits from the measured speed fluctuations was carried out. It was expected that by using the speed fluctuations for the velocity as well as acceleration normalization and valve acceleration computation, the results would be more precise.

Finally, we carried out a comparison of the measured data of the half-engine setup and the partial-engine setup. The half-engine setup utilizes nearly a full-engine assembly. The partial-engine setup, on the other hand, utilizes only the head and the head cover and the related valvetrain components. This setup is commonly used in the automotive industry to simplify valvetrain tests. It was expected to deliver values close to the half-engine setup, although no direct comparison has been published. The results show a wide variety of different analyses that can be performed from a single measurement.

Keywords: valvetrain, speckle noise, signal drop-out, laser Doppler vibrometry, USS protocol, data acquisition, automation

Abstrakt

Tato práce popisuje návrh a vývoj nové generace testovacího zařízení pro automatizované měření dynamické odezvy ventilových rozvodů a předkládá analýzy naměřených dat. Zaměřuje se na vývoj měřícího softwaru a algoritmů pro detekci výpadků dopplerovského signálu jakožto klíčových částí měřícího řetězce, které umožnily toto měření plně automatizovat. V konečném důsledku byl čas měření významně zkrácen, a tudíž byla zvýšena efektivita testovacího zařízení a opakovatelnost měření. Zařízení je koncipováno jako mnohoúčelové a použitelné i v jiných odvětvích měření spalovacích motorů.

V textu práce je popsána naše implementace USS protokolu do LabVIEW, jež byla dlouhodobě nevyřešena a zároveň velmi žádána. Protokol je použit jako rozhraní ke komunikaci s frekvenčním měničem Siemens. Frekvenční měnič pak řídí elektromotor, který pohání spalovací motor při měřeních. V práci jsou rovněž zmíněny problémy, které se při implementaci protokolu objevily a jejich řešení.

Podstatná část práce popisuje algoritmus pro identifikaci výpadků signálu, který byl vyvinut pro měření kinematiky ventilů pomocí laserového dopplerovského vibrometru. Tyto výpadky signálu se objevují při nízké amplitudě dopplerovského signálu jakožto důsledek dynamických změn speckle šumu během rotace a náklonu ventilu. Detekční algoritmus ukládána pouze data nepostihnutá tímto jevem, což výrazně snižuje čas nutný k následné analýze naměřených dat. V práci jsou uvedeny nejčastější typy výpadků, jejich vlastnosti a umístění. Jsou popsány vytvořené algoritmy sloužící k automatické separaci měřeného signálu. Toto umožňuje přesnější zacílení parametrů testů rozpoznávajících výpadky v daném intervalu (otevírání ventilu, zavírání ventilu atd.).

K zajištění automatizace, bezpečnosti a jisté inteligence vytvořeného systému bylo nutné kombinovat pokročilé techniky pro paralelní sběr dat, komunikaci s vícero přístroji a zpracování dat.

K ověření schopností systému byly proměřeny dva OHC ventilové rozvody – nejprve standardní sériový ŠKODA 1.2 HTP a následně lehkoběžná verze tohoto rozvodu. Výsledky obou rozvodů byly porovnány v několika detailních analýzách. Bylo rovněž provedeno porovnání dvou přístupů k výpočtu zrychlení ventilu z naměřených dat rychlosti. Běžně je pro tyto účely zanedbávána nerovnoměrnost otáčení vačkového hřídele. Jelikož náš systém je schopen měřit průběh rychlosti otáčení hřídele, bylo toto do výpočtu zahrnuto. Předpokládalo se, že tato informace zpřesní výsledek výpočtu.

Na závěr bylo provedeno měření tzv. částečného uspořádání motoru (partial-engine setup). Naměřené kinematické veličiny ventilů byly porovnány vůči hodnotám z předchozího měření na maketě motoru (half-engine setup), která se skládá z téměř úplného motoru. Naproti tomu částečné uspořádání využívá pouze hlavu válců, víko a nezbytně související části rozvodu. Toto uspořádání je běžně používáno v automobilovém průmyslu k zjednodušení měření rozvodů. Ačkoli se předpokládá, že poskytuje měřené hodnoty blízké hodnotám z měření na maketě, nebylo přímé porovnání nikdy publikováno (s ohledem na znalosti autora). Výsledky ukazují širokou škálu analýz, jež mohou být provedeny z jednoho měření.

Klíčová slova: ventilový rozvod, speckle, výpadky signálu, laserová dopplerovská vibrometrie, USS protokol, sběr dat, automatizace

Acknowledgement

The work reported in this thesis was carried out at the Technical University of Liberec, The Institute of Novel Technologies and Applied Informatics from 2006 to 2011.

This research was done in collaboration with ŠKODA AUTO a.s and LENAM, s.r.o. I wish to thank Ing. Jan Lohniský and Ing. Milan Pipal for their valuable insights, assistance in the measurements, the great discussions and feedback.

Thanks also to Mgr. Jan Březina, Ph.D. for the fruitful discussions concerning paper III and thanks to Ing. Petr Horník for the considerable amount of mechanical work he did while preparing the engines for measurements.

Big thanks to my Kendra Hoover for revising the (non)English of the manuscript, being patient with my sometimes late arrivals home and being patient with me in general.

Finally, I would like to thank my family and all my friends for their support and encouragement during this project.

Liberec, December 2011 Petr Hošek

List of terms, symbols, and abbreviation

AE	Action Engine
AI	Analog input
AO	Analog output
ASIC	Application-specific integrated circuit
BCC	Block Check Character
BICO	Binector Connector Technology
BOP	Basic Operator Panel
CDS	Command Data Set
DAQ	Data acquisition
DBL	Double precision type
DDS	Drive Data Set
DMA	Direct memory access
DOHC	Double overhead camshaft, valvetrain design type
DWT	Discrete Wavelet transform
Out_1	First exhaust valve
HIW	Actual value, from German *Hauptistwert*, USS protocol parameter
HSW	Main setpoint, from German *Hauptsollwert*, USS protocol parameter
I/O	Input/output
IC engine	Internal combustion engine
In_1	First intake valve
IRC	Incremental rotary encoder
LGE	from German *Telegrammlänge*, USS protocol parameter
LDV	High speed Laser Doppler Vibrometry
LVOOP	LabVIEW object-oriented programming
OHC	Overhead camshaft, valvetrain design type
OHV	Overhead valve, valvetrain design type
PFI	Programmable Function Input
PKW	from German *Parameter-Kennung-Wert*, USS protocol parameter
Ppm	Parts-per-million
Ppr	Pulse per revolution, resolution of the IRC sensor
PWM	Pulse-width modulation
PXI	PC-based platform for test, measurement, and control
PZD	from German *Prozeßdaten*, USS protocol parameter
QSM	Queued state machine
Rpm	Revolutions per minute
SM	State machine
STC	System Timing Controller
STW	Control word, from German *Steuerwort*, USS protocol parameter
STX	Start of text, control character (0×02)
TTL	Transistor–transistor logic
TUL	Technical University of Liberec
USR	Un-initialized Shift Register
USS	*Universelles serielles Schnittstellen-Protokoll,* Universal Serial Interface Protocol
Vi	Virtual instruments, LabVIEW data file type
ZSW	Status word, from German *Zustandswort*, USS protocol parameter

Err	Maximal frequency measurement error for particular method
a	Valve acceleration
a_n	Normalized valve acceleration
f	AC power frequency
f_{cam}	Frequency camshaft
f_{crank}	Frequency of crankshaft
f_k	Frequency of known source signal, e.g. timebase
f_p	Frequency of IRC pulses
f_s	Center frequency of carrier signal generated by Bragg cell
f_x	Frequency of unknown signal, e.g. signal to be measured
N	Divide down value, divider
n	Preselected amount of revolutions
n_r	The speed of rotor of asynchronous motor
n_s	The synchronous speed of the AC motors
p	Number of pole pairs; parameter of algorithm
rpm_{cam}	Revolutions per minute of camshaft
rpm_{crank}	Revolutions per minute of crankshaft
$rpm_{cam\ ref}$	Smoothed data from speed fluctuation measurement
$rpm_{cam\ meas}$	Unprocessed data from speed fluctuation measurement
$\overline{rpm}_{cam\ ref}$	Mean of $rpm_{cam\ ref}$
s	Asynchronous motor slip; valve lift
t	Time
v	Valve velocity
v_n	Normalized valve velocity
$\Delta f(x)$	Forward difference
Δs_{max}	Difference in maximal valve lift
α	Angle between the illumination and the backscattered light
φ	Shaft angle
λ	Central wavelength of the emitted light
ω	Angular frequency (of shaft)

List of original papers

I P. Hosek, T. Prykäri, E. Alarousu and R. Myllylä, Application of LabVIEW: Complex Software Controlling of System for Optical Coherence Tomography Using LabVIEW, Journal of the Association for Laboratory Automation, April 2009, Volume 14, Issue 2, pp. 59-68.
The software design patterns described in this paper were used to create the final application for valvetrain measurements developed as a part of this thesis.

II P. Hosek, M. Diblik: Implementation of Siemens USS Protocol into LabVIEW, Journal of the Association for Laboratory Automation, October 2011, Vol. 16, Issue 5, pp. 347-354

III P. Hosek, Algorithm for signal drop-out recognition in IC engine valve kinematics signal measured by laser Doppler vibrometer, Journal of Optics & Laser Technology, DOI: 10.1016/j.optlastec.2011.09.034, accepted 9/2011

IV P. Hosek, Comparison of camshaft speed fluctuations and valvetrain kinematics of engine in half setup and partial setup, *in progress (chapter 11 of the thesis)*

Contents

1 Introduction

1.1 Background and motivation of the work

The internal combustion (IC) engine has been evolving for almost one and half centuries (1866, Daimler, Benz, first car with combustion engine). Nevertheless, the keystones still remain the same. We still use cams and valves while the research focused on replacing those parts with electromagnetic elements is in progress.

In the electromagnetic actuation concept, the opening and closing of the valve is obtained by alternatively energizing upper and lower magnets with an armature connected to the valve. This actuating principle offers maximum flexibility and dynamic response in valve control, but despite a decade of significant development efforts, the main drawbacks of the concept (it being not fail-safe and its high energy absorption) have not been fully overcome [1]. Thus those elements remain electromechanical [2] or hydraulically-actuated [3] as in case of the MultiAir Technology from Fiat [4].

Testing is an important part of the design process of each individual component. First, it enables checking of the correct functionality while at the same time monitoring the critical values that might negatively influence the lifetime of the final product. The measurement of the valve displacement and velocity is a fundamental part of the validation of the valvetrain design. With high speeds the eigenfrequency and stiffness of each part, the mass distribution, resonance or friction begin to play an important role that may lead to changed cylinder fluid dynamics. Phenomena like valve float and valve bounce may appear. Valve float can be observed when the inertial force of the valvetrain components exceeds the spring force of the valve springs, thus allowing components to separate. In addition, valve float causes the valve to exceed the maximum lift of the kinematic motion and close with an abnormally high velocity [5]. Valve bounce occurs when the valve closes against the seat with such a high velocity that it physically bounces off the seat and remains open as the piston begins the compression cycle [5], [6]. If the valve is repeatedly seated with too high velocity (so called impact velocity [7]) the seat can be worn off. Contact fatigue or other damage may appear. Due to resulting leakage, valve heat dissipation is not ensured and local heat load increases. The valve can even melt. To prevent or minimize those effects we have to have an apparatus to discover the conditions of their occurrence.

High speed Laser Doppler Vibrometry (LDV) has become a standard measurement technique for obtaining the kinematics of the valves [6], [8]. The main reason is that the technique is non-contact and offers information about both the valve displacement (using the fringe counting technique [9], [10]) and the valve velocity (based on the Doppler effect) up to high engine speeds. The LDV technique also has its drawbacks. It is sensitive to dust and oil droplets that might appear during the measurement and demands precise focusing of the lenses. However, the main problems arising during the automation of the measurement are *speckle noise* and *signal drop-outs* [11], [12], [13].

The current measurement procedure of the valve kinematics is time consuming. The measurement has to be carried out over the given (operational) spectrum of the engine rpm. The higher frequencies tend to be more interesting for further processing so they are measured in detail with smaller rpm increment. Ideally all the valves (all cylinders, exhaust and intake valves) are examined. It might take a day to measure the response of a valvetrain with newly designed curvature of a cam or when new valve springs are used. To save time, usually only the valves closest and endmost to the camshaft drive are examined.

In industry it is necessary to have high throughput of the testbed, which results in inability to check the occurrence of the drop-out phenomenon in detail. This is partly because of the amount of the measured data and partly because some problematic parts can be discovered only after a very detail processing.

A fully-automated system which would significantly reduce the time needed for the complete analysis of the valvetrain components would be a great tool. Such a system has to offer prompt feedback for the design engineers, repeat measurement in case of recognition of signal drop-out and deliver as much information as possible from a single measurement.

1.2 Contribution of the thesis

To test and analyze the dynamics of the valvetrain is essential since it presents the part of the engine where even slight changes directly influence the engine setting, lifetime and fuel consumption [14].

The measurement system we designed and constructed can be called a new generation of the apparatus commonly used for the valve kinematics measurements [6], [8]. The systems described in the referenced papers are rather experimental and have their limitations. On the other hand, our apparatus is aimed at industry. It spans all of the engine's operational speeds and can operate with a half-engine or a partial-engine setup (partial-engine setup utilizes only the head and the head cover and the related valvetrain components). As described later, it also automatically compensates for slippage of the ribbed belt that is used to transfer the torque between the shaft of the electromotor and the combustion engine. Moreover, it is capable of a truly parallel measurement of the camshaft speed fluctuations, which can help to discover a faulty valvetrain component or to improve the precision of calculations that presume the speed of the shaft to be constant. Most of all, with the algorithm for LDV drop-out noise detection, the system is fully automated. It runs the measurement across predefined engine speeds, repeating if drop-out noise is detected and saving only the representative data for further processing.

With the constructed system, we carried out a measurement of the valvetrain kinematics and the camshaft speed fluctuations of engine in half setup and partial setup. The partial-engine setup is commonly used to make the measurement faster and cheaper, although no direct comparison of the measurement results has been presented before.

The work is based on cooperation with ŠKODA AUTO a.s. The created system is planned to be used for several educational purposes during experimental lectures at TUL. The results of this work are going to be applied in the automotive laboratories of ŠKODA AUTO a.s.

1.3 Content of this work

In this thesis project, we focused on the design and construction of the apparatus for automatic measurement of valvetrain kinematics and speed fluctuations. We carried out numerous measurements to demonstrate its overall capabilities and to compare different valvetrains and engine setups.

In Chapter 2, valvetrain designs that are nowadays used are briefly described. The basic terminology is stated and the valvetrain components are depicted.

Chapter 3 presents the designed apparatus while showing its schema and discussing the specifications of each part.

Chapter 4, which was published as a separate article (paper II), focuses on the development of the USS protocol communication library for driving the Siemens frequency inverters.

Chapter 5 describes the frequency measurement techniques which use the programmable counters/timers of the counter board. The error estimation of each method is calculated and the most suitable method for IC engine shaft fluctuations measurement is chosen. The error calculations are compared to a real measured IRC signal.

Chapter 6 introduces two additional tasks that utilize the programmable counters. The first routine enables automatic pulse per revolution (ppr) identification. The second one serves to delay the trigger pulse to always obtain a data record that starts on the base circle of the cam and spans the whole phase of the valve opening and closing. This routine prevents the need to manually reposition the IRC sensor when moving from one valve to another.

Chapter 7 focuses on crucial graphical-programming architecture patterns. They can be understood as keystones for any mid-sized real-time measurement application (published paper I). Chapter 8 then shows those generic patterns combined together into software for the measurement of valvetrain kinematics and shaft speed fluctuations.

Chapter 9 presents algorithm for detection of signal drop-outs in laser Doppler signal. The work shows the most commonly encountered drop-outs during the valve kinematics measurements and their characteristics and locations. It presents an automatic separation algorithm for the measured signal so that the drop-out recognition tests can be aimed at specific data intervals (valve opening, valve closing, etc.) with specifically set parameters of the algorithm (paper III).

To verify the capabilities of the designed apparatus, two different valvetrains in half-engine setup were measured. Analyses of the measured data can be found in Chapter 10.

Chapter 11 presents the results of the comparison of the measured data of the half-engine setup and the partial setup. It poses the first direct comparison of these two engine setups (paper IV).

The results of the work and future research are discussed in Chapter 12.

2 Valvetrain

Before we describe the designed apparatus, developed software system and the measurement techniques used it is necessary to familiarize with the basic terminology of valvetrains since in the following text some elements will be referenced. More detailed description of the valvetrain designs can be found in [14]. Detailed valvetrain dynamics description can be found in [15].

Valvetrain is mechanism which ensures opening and closing of the valves and thus controls the air and fuel flow into and out of the cylinders, facilitating combustion. Parts of the valvetrain, depending on the construction, are typically camshaft with gear, lifters, pushrods, rocker arms, springs with accessories and valves (Fig. 2.1). According to the technical design rocker arms and pushrods can be omitted. In some (atypical) cases even return springs. Valvetrains are built in several configurations but the most common in combustion engines are the OHV and OHC (DOHC) layouts.

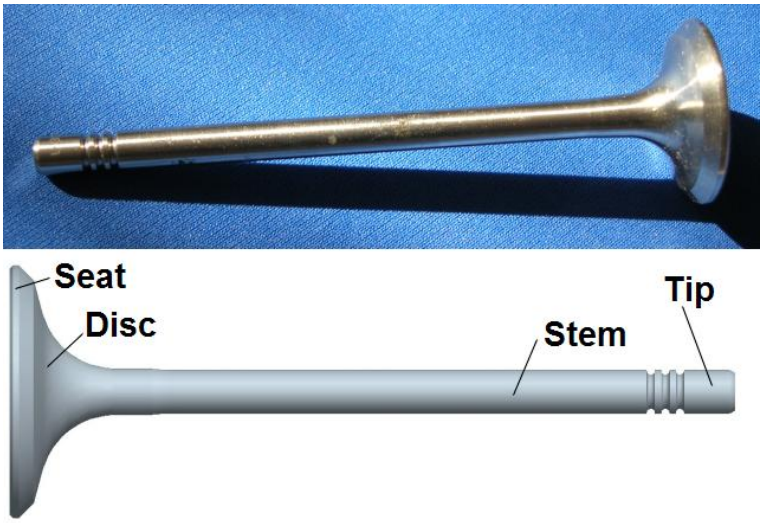


Figure 2.1 – Valve

2.1 OHV

The valvetrain design that is nowadays overcome by more modern OHC. An overhead valve internal system is one in which the intake and exhaust valves and other parts are contained in the cylinder head. As a rule, the camshaft is placed in the cylinder block close to the crankshaft (Fig. 2.2) and uses pushrods to actuate rocker arms above the cylinder head to actuate the valves. The drive is done by short chain or by gearing.

The main advantage is cheaper cast of the cylinder head, which is simpler than in the OHC, thus when servicing it is not necessary to trouble with the adjustment of the position of the camshaft against the crankshaft. The valve clearance can be adjusted via the screw on the rocker arm although the latest OHV engines don't need the regular adjustment of the clearance between the valve stem and the rocker arm since this is done automatically thanks to hydraulic lash adjusters.

The pushrod engines have more valvetrain moving parts thus more valvetrain inertia and mass, therefore, pushrod engine cannot usually revolve at speeds as high as OHC designs. Also to control four valves per cylinder is more demanding.

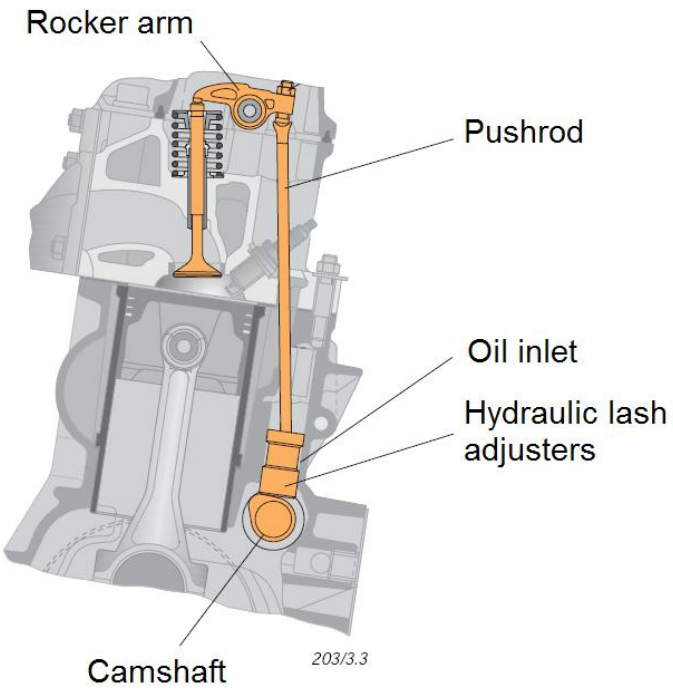


Figure 2.2 – OHV valvetrain 1.4l/44kW [16]

2.2 OHC (DOHC)

The modern type of valvetrain used nowadays in most of the combustion engines. The abbreviation means Overhead Camshaft and reveals that the engine camshaft is placed within the cylinder head, above the combustion chambers, and drives the valves or lifters in a more direct manner compared to OHV. The drive is assured with a timing chain, timing belt or rarely with a set of wheels. The valves can be actuated by the mean of rocker arms (Fig. 2.3) or directly through bucket tappets (Fig. 2.4). The mass of the moving parts is then minimal possible and in addition the parts can be light thus the spring force can have lesser value. One of the key components is the hydraulic lash adjuster that is used to maintain zero valve clearance. The design of the hydraulic lash adjuster used with rocker arms is depicted in Figure 2.5.

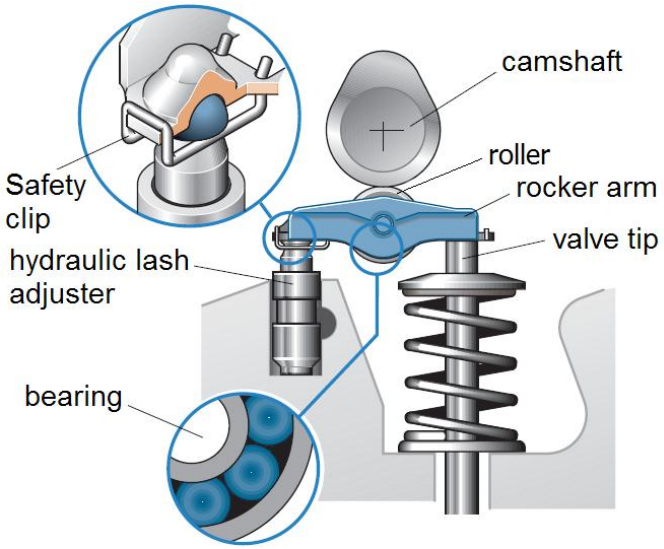


Figure 2.3 - The valve actuation through the rocker arm [17]

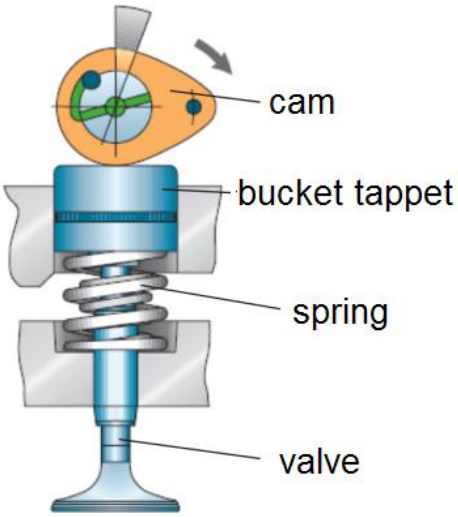


Figure 2.4 – The valve actuation through the bucket tappet [18]

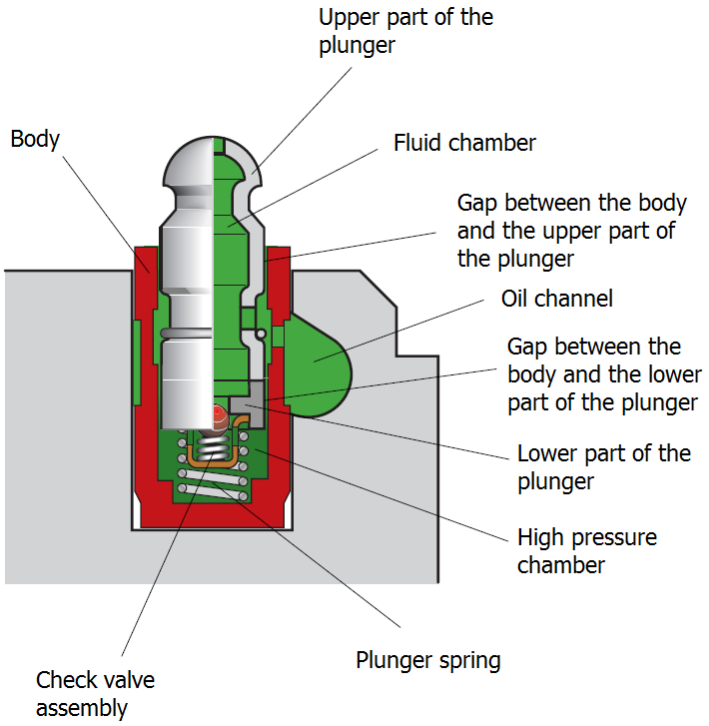


Figure 2.5 – Hydraulic lash adjuster [17]

The advantages and disadvantages are opposite to OHV valvetrain. In addition double mechanism for exchanging the content of the combustion chamber can be used. One handles the intake and the other the exhaust. This design is called DOHC (2xOHC) and can also use either rocker arms (Fig. 2.6) or bucket tappets (Fig. 2.7).

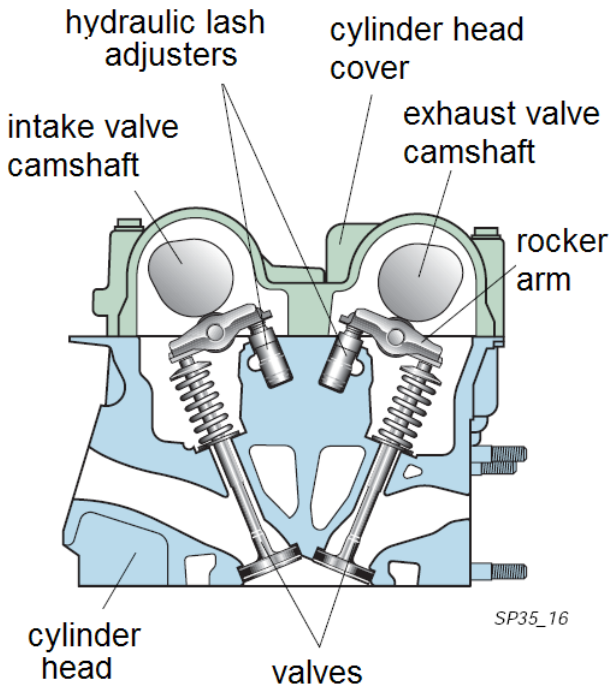


Figure 2.6 – DOHC, rocker arms [17]

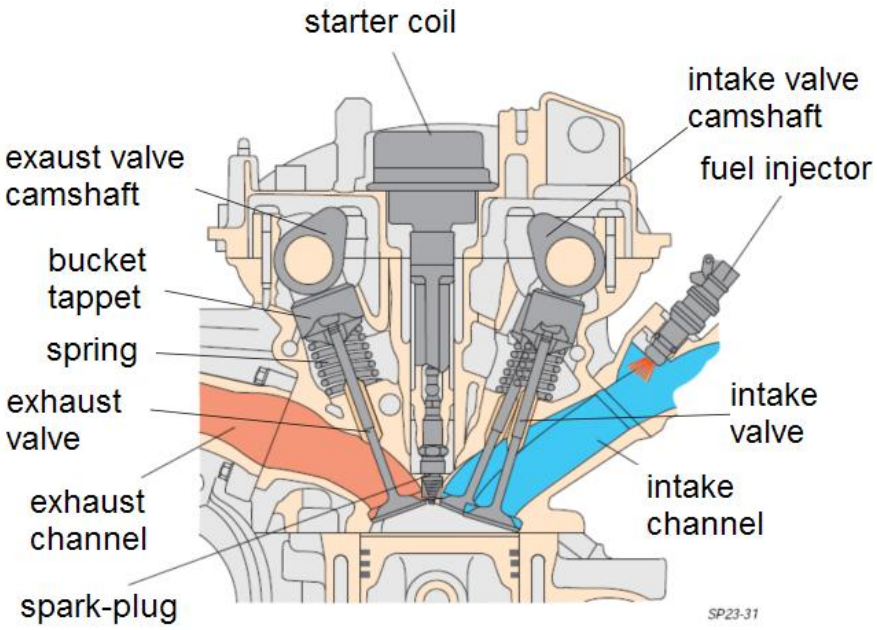


Figure 2.7 – DOHC, bucket tappets [19]

3 Design of the measurement apparatus

The apparatus for valvetrain testing is a very complex system spanning many different technical fields. It is designed to reproduce the functional conditions in cars.

At the beginning we have a combustion engine with a set up valvetrain and all the accessories so the dynamics of the valvetrain is preserved. When the measurement is finished we should have the desired characteristics (displacement and velocity of the valve) of the valvetrain under examination and shaft speed fluctuation curve. How to accomplish that will be described in the next paragraphs, where we will state individual basic parts of the measurement chain and the criteria of their selection. The schema of the new apparatus will be presented.

3.1 Developed measurement system – the hardware

The measurement system (Fig. 3.1, Fig. 3.2) utilizes electromotor and frequency inverter with communication unit to drive it. The electromotor is used to drive the crankshaft of the combustion engine. Between the electromotor and the crankshaft the step-up gear is installed to allow the full range of measurement rpm, typically $rpm_{crank} = 0 – 6000$ rpm. The camshafts (DOHC) in the most common setup are connected to the crankshaft via a timing chain. The incremental encoder (IRC) is placed either at the end of the intake valve camshaft or the exhaust valve camshaft offering the information about the camshaft displacement and subsequently about the camshaft speed fluctuations and the actual speed of rotation. The cylinder head cover had to be modified to allow that. If needed, the IRC can also be mounted at the end of the crankshaft. The camshafts of the test engine were driven by the crankshaft using a timing chain with a reduction ratio of 1:2.

Apertures were milled into the place where pistons normally belong. The pistons were removed and in the created space the laser Doppler probes were installed. The original crankshaft was replaced with a modified straight ‘dummy’ shaft, only to drive the connected valvetrain and the oil pump. No gas forces or combustion forces occur in the measurement system. Excluding those forces, however, does not compromise the validity of the experimental data [20], [8]. In this thesis, we will refer to the assembly as a *half-engine setup*.

The measured valve is always equipped with retro-reflective tape which is placed at the center of the bottom side of the plate. One fiber optics laser probe is aimed at the tape and monitors the motion of the valve (measurement arm). It works as a transmitter and receiver at the same time. The second probe is aimed at the cylinder head (reference arm) itself which is also rigged with the tape. The probes are connected to a differential laser Doppler system which uses Mach-Zehnder interferometer design. The vibrations of the cylinder head arising during the measurement are subtracted from the motion of the valve. The measurement is thus more accurate. The connected laser vibrometer controller handles the evaluation of the incoming signals and outputs analog signals representing the displacement and velocity of the monitored valve. Those signals are connected to two channels of the data acquisition (DAQ) card. Only the component of motion along the optical axis of the instrument can be determined, so the two beams must be aligned parallel to the valve stem to have a correct measure [8].

The pulse signal from the IRC encoder is used as an external sampling source so the data are sampled depending on the actual cam displacement. The standard for this type of measurement is 720 pulses per revolution (ppr). This offers resolution 0.5° of the camshaft which is fine enough, preserves good readability, shows step changes in acceleration that can

discover problematic parts and doesn't pose a problem of handling huge amount of data. We also use higher resolutions (i.e. 1800 and 3600 ppr). Moreover, the IRC signal is used by the controlling software to compensate slippage of the ribbed belt which is used to transfer the torque between the shaft of the electromotor and the combustion engine. In addition, the IRC enables monitoring shaft speed fluctuations during one revolution of the camshaft. The reference signal (one impulse per rotation) serves as a trigger signal to start the data acquisition and reset the interferometer.

The main parts of the apparatus are embedded into a frame that was specially built for our purposes offering stability, rigidity, and minimizing the vibrations and the impact on the surrounding environment. It can hold common three cylinder and also four cylinder engines.

We also installed a system for oil heating and cooling which allows setting and monitoring of desired operating temperature as well as pressure sensing. The whole measurement is operated by developed software.

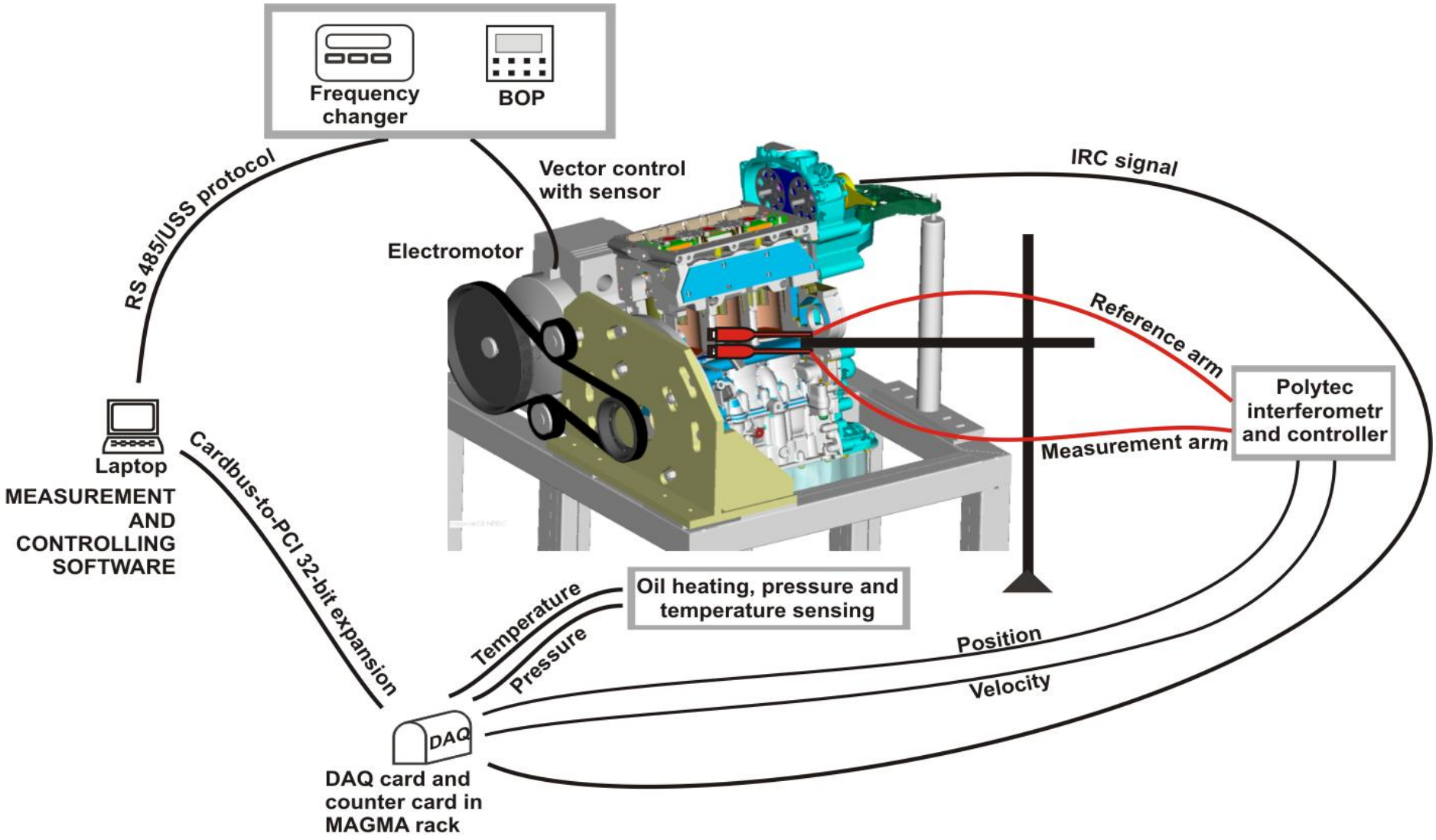


Figure 3.1 – Schema of apparatus for valvetrain measurements

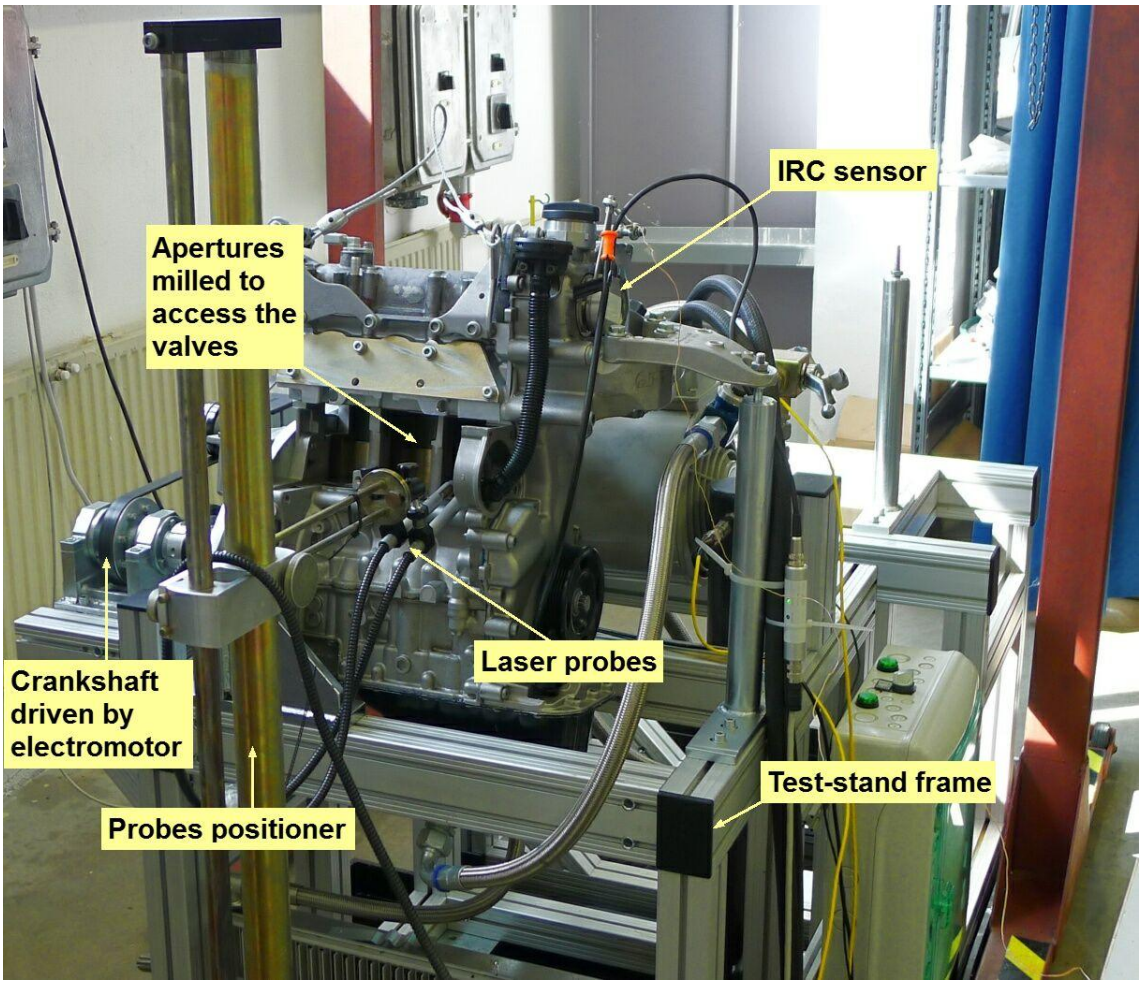


Figure 3.2 – Photo of the apparatus in half-engine setup

3.1.1 Combustion engine

We carried out our measurements on the ŠKODA AUTO three cylinder 1.2-litre 12-valve combustion engine that can nowadays be found in the basic models of Škoda Fabia, Škoda Roomster or VW Polo offering the maximal power of 47 kW (64 bhp) at 5400 rpm.

3.1.2 Electromotor, inverter, control unit, gearbox

The selection of the electromotor was crucial. We needed an electromotor that would be capable of driving the three cylinder engine, as well as more powerful four cylinder engines for future purposes and experiments. It had to be capable of starting the revolution of a cold engine (oil has low viscosity) and suitable for driving either the crankshaft, or the camshaft directly. Our demands were the following:

- Crankshaft rpm range 1000 – 8500/min, if needed with the assistance of gearing
- Nominal torque ~10 Nm
- Starting torque 30++ Nm
- High reliability, maintenance-free

The disadvantages of the brushed DC motor made this motor type unsuitable for our purposes. The main drawbacks are frequent maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the brushes and springs, which carry the electric current as well as cleaning or replacing the commutator [21].

The brushless DC motors, however, offer long life-span, little or no maintenance, and high efficiency but include high initial cost and more complicated motor speed controllers [21].

It would be possible to choose an AC motor that provides full rpm range, but such a solution would be very expensive. Instead, we have chosen the asynchronous AC Siemens 7.5 kW electromotor with the following parameters (Table 3.1) and gearing 2:1. The ribbed belt is used to transfer the torque between the shaft of the electromotor and the combustion engine, which ensures that in critical situations (such as engine jamming) the belt will be allowed to slip, thereby not causing more damage to the engine.

Table 3.1 – Electromotor parameters

Type	1LA7131-2AA10-ZH57
Rated power	7,50 kW
Rated torque	24,40 Nm
Starting torque	56,12 Nm
Rated current	13,80 A
Rated rpm	2930 rpm
Rated voltage	400 V
Number of poles	2

The asynchronous induction motor is by far the most widely used choice for development application in industry. Being both rugged and reliable, it is also the preferred choice for the variable-speed drive applications; offering low cost, high reliability and fairly high efficiency.

The frequency inverter Siemens Sinamics G120 with communication unit CU 240S is used to drive the electromotor. In combination with a quadrature encoder attached to the shaft of the electromotor, it regulates the rpm of the electromotor. The synchronous speed n_s of the AC motors is given by

$$n_s = \frac{f \cdot 60}{p} \tag{3.1}$$

where f is the AC power frequency (Hz) and p is the number of pole pairs. The speed of the rotor n_r is given by

$$n_r = n_s(1 - s) \tag{3.2}$$

in the case of asynchronous motor (where s is the slip).

The key issue in speed control of the induction motor is the frequency of the source. Variable Speed Drives (in our case the Sinamics inverter) can produce variable frequency. The usual method used for adjusting the motor voltage is pulse-width modulation (PWM). With PWM voltage control, the inverter switches are used to divide the quasi-sinusoidal output waveform into a series of narrow voltage pulses and modulate the width of the pulses. This way, the motor speed is proportional to the applied frequency as can be seen from the equation (3.1).

During the design of the measurement apparatus, we also had to decide what communication bus and protocol to use. As we were developing a superior computer-based system, it was necessary to choose such an interface that would offer simple but reliable

communication between the electromotor and the PC. Siemens offers a wide spectrum of connection possibilities of their control units that can operate the SINAMICS G120 inverter, including PROFIBUS (RS485), PROFINET (Ethernet), or USS protocol (RS485). As we have just a simple topology PC (master) – Inverter (slave), we have chosen the standard version of the control unit (CU 240S) that allows us to connect over RS485 and communicates via the USS protocol.

We also installed an additional cooling fan to prevent overheating when the desired rpm is low, and thus the self-cooling capabilities of the electromotor might be insufficient.

3.1.3 Incremental rotary encoder (IRC)

Another key component was the incremental rotary encoder (IRC). Demands for the sensor were quite high:

- Shaft speed range 0÷8500/min
- Adjustable resolution from 0.5° to 0.1°
- Adequate accuracy
- TTL signal
- High vibration resistivity, robustness

We considered using the magnetic type of IRC whose simple noncontact design makes this encoder technology low cost and reliable. Nevertheless, this type doesn't offer high accuracy with high speeds and adjustable resolution in the same time making it unsuitable for our application.

We decided to use Kistler shaft encoder set which is designed for measurements on combustion engines and works on optical principle. Following table shows its properties:

Table 3.2 – IRC technical data, Kistler 2614A

TTL crank angle signal Resolution	°	0.1 ... 6
Dynamic accuracy at 10,000 1/min (signal delay)	°	0,02
TTL trigger signal (TRG) resolution	°	0.1 ... 6
Speed range	1/min	1 ... 20'000
Operating temperature range, Encoder and amplifier	°C	-30 ... 60
Connection flange	°C	-30 ... 100
Power supply with stabilized voltage	VDC	5 ±0,25
Current consumption	mA	200
with unstablilized voltage	VDC	6 ... 24
Current consumption	mA	200 ... 400
Mounting diameter of encoder	mm	69
Encoder weight	g	460
Amplifier dimensions	mm	98 x 64 x 37
Amplifier weight	g	300

3.1.4 Data acquisition card, counter card

Since the superior software that serves for full automation, control and processing of the measurement was planned to be developed in LabVIEW to obtain the full support and benefits of using software and hardware of one family we purchased the National Instruments data acquisition card. We needed device that would offer:

- 2 or 4 analog inputs; dedicated A/D converter per channel
- 16-bit input resolution
- Sampling rate 1 MS/s/ch
- Mobility, robustness

We meant to use the DAQ card in combination with laptop to achieve high mobility and usability of the card also for other measurements. Since our demands were quite high we couldn't find any sufficient device for USB or PCMCI slot and PXI system was too expensive. The only possible solution was to purchase external PCI expansion chassis and connect it with the laptop via ExpressCard module. The DAQ card then appears as it was present in the laptop itself. For that reason we ordered Magma CB264 2 slot PCI expansion chassis with combination of NI PCI 6120 DAQ card and BNC 2110 connector. The card parameters are following:

Table 3.3 – NI PCI 6120 DAQ card technical data [22], [23]

Bus	PCI
Analog inputs	4 - dedicated A/D converter per channel
Input resolution	16-bit
Sampling rate	1 MS/s
Input range	±0.2 to ±42 V
Analog outputs	2
Counters/Timers	2, 24-bit
Triggers	Analog, digital

For the camshaft speed fluctuations measurement we needed programmable counters. For further analyses we wanted to have the valve displacement and velocity samples synchronized with the actual speed of the revolution. For that reason we couldn't use the counters on the DAQ board itself since they are not triggerable. This means that the frequency reading wouldn't start with the reference mark of the IRC sensor but would be software triggered. Therefore we utilized a separate counter board NI PCI 6602 + BNC 2121. The card offers 8 triggerable counters and 80 MHz oscillator that contributes for more accurate frequency reading (the PCI 6120 has 20 MHz clock). The amount of the counters available also allowed us to perform additional tasks during measurement.

Table 3.4 – NI PCI 6602 counter card technical data

Bus	PCI
Counters/Timers	8 up/down, 32-bit
Source frequency	80 MHz
I/O lines	Up to 32 digital I/O lines (5 V/TTL)

3.1.5 Laser probes

The noncontact measurement of valve motion is performed by a vibrometer system (Fiber Interferometer Polytec OFV 502 + vibrometer controller Polytec OFV 3000) that is comprised of an optical sensor head and a controller which provides power for the measuring head and processes the vibration signal. This signal is decoded in the controller to achieve velocity and displacement information [24].

A laser Doppler vibrometer is in principle a velocity sensor. The Doppler shift from the vibrating object modulates a carrier signal at a center frequency of f_s = 40 MHz generated by the Bragg cell in the optical head. The purpose of the velocity-decoding electronics is to

convert the frequency seen by the photo detector into a voltage proportional to the vibration velocity *v*. The relation between the measured frequency and the velocity is given by [25], [26]:

$$f_D = f_s + 2\frac{v}{\lambda} sin\frac{\alpha}{2} \tag{3.3}$$

Where f_D is the Doppler frequency, f_s is the frequency of the carrier signal, λ is the central wavelength of the emitted light, v is the component of the velocity of the moving object, α is the angle between the illumination and the backscattered light received by the probes.

The displacement is derived independently. The displacement decoder utilizes a digital method called fringe counting [9], where the number of fractional wavelengths is counted while the surface moves.

Table 3.5 – Vibrometer parameters

Velocity Ranges	5, 25, 125 and 1000 mm/s/V, (+/- 10 m/s full scale)
Velocity Resolution	0.5 um/s
Displacement Ranges	80, 320, 1280 and 5120 µm/V, (+/- 51.2 mm full scale)
Displacement Resolution	0.08 um
Bandwidth	DC to 1.5 MHz
Velocity Filter Steps	5k, 20k, 100k, 1.5M Hz (-3dB)
Velocity Tracking Filter	Slow, Fast or Off

4 Implementation of Siemens USS Protocol into LabVIEW

We developed the measurement and control application in LabVIEW development environment, so the essential question became the implementation of the USS protocol to communicate with the frequency inverter to drive the electromotor, as no LabVIEW command libraries were available. Of course, there was an option to use the OPC server either from Siemens or third party, but it required extra money, installation, and setting of another application and driver not mentioning always the best performance and the necessity to resolve compatibility issues. There was also permanent demand from the community of LabVIEW users to have native LabVIEW implementation of the USS protocol. Based on above written, we decided to write collection of the most common USS commands. The final library was published for public use on National Instruments discussion forum – Motion Control and Motor Drives board.

4.1 Basic protocol overview

The USS protocol (Universal Serial Interface Protocol) defines an access technique according to the master – slave principle for communication via a serial bus. This also includes the point-to-point connection. Essential features of the USS protocol are as follows:

- It supports a multi-point-capable coupling, e.g. EIA RS 485 hardware
- Master-slave access technique
- Single master system
- Max. 32 nodes (max. 31 slaves)
- Simple, reliable telegram frames
- Easy to implement
- Operation with either variable or fixed telegram lengths

For detail protocol specifications see [27].

4.2 Structure of USS telegram

4.2.1 Description

Figure 4.1 shows the structure of a typical USS telegram.

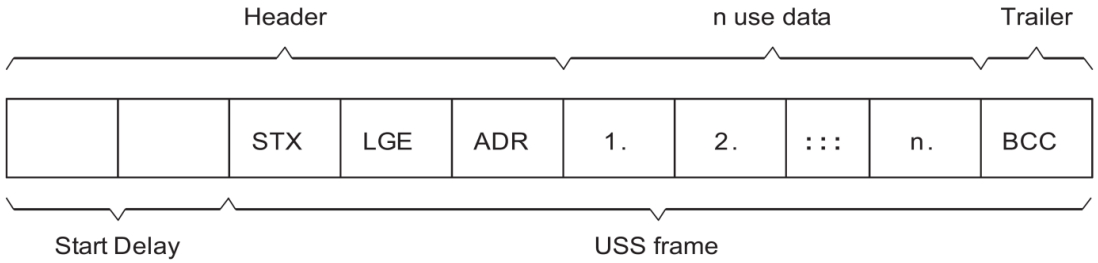


Figure 4.1 – Structure of USS telegram [28]

Variable length telegrams and fixed length telegrams can both be used. This can be selected using parameters P2012 and P2013 to define the PZD (from German *Prozeßdaten*) and PKW (from German *Parameter-Kennung-Wert*) lengths which are the two parts of the *use data block*. The fixed length sizes we used are shown below:

STX		1 byte
LGE		1 byte
ADR		1 byte
Use data:	PKW	4 words = 8 bytes
	PZD	8 words = 16 bytes
BCC		1 byte

SUM		28 bytes

4.2.2 STX

The STX (*Start of text*) field is a single byte ASCII STX character (0x02) used to indicate the start of a message.

4.2.3 LGE

The LGE (from German *Telegrammlänge*) is a single byte field, indicating the number of bytes which follow this in the telegram. It is defined as the sum of:

- use data characters (quantity n = 24 in our case)
- address byte (ADR)
- block check character (BCC)

The actual total telegram length will of course be two bytes longer as STX and LGE itself are not counted in the LGE.

4.2.4 ADR

The ADR field is a single byte containing the address of the slave node (e.g. inverter) and additional telegram settings (e.g. broadcast, mirror telegram).

4.2.5 BCC

BCC means block check character. It is an exclusive OR (XOR) checksum over all telegram bytes except the BCC itself.

4.2.6 Structure of use data block

The use data area of the USS protocol is used for transferring the application data, which are the earlier-mentioned parameter channel PKW and process data-channel PZD as can be seen in the Figure 4.2.

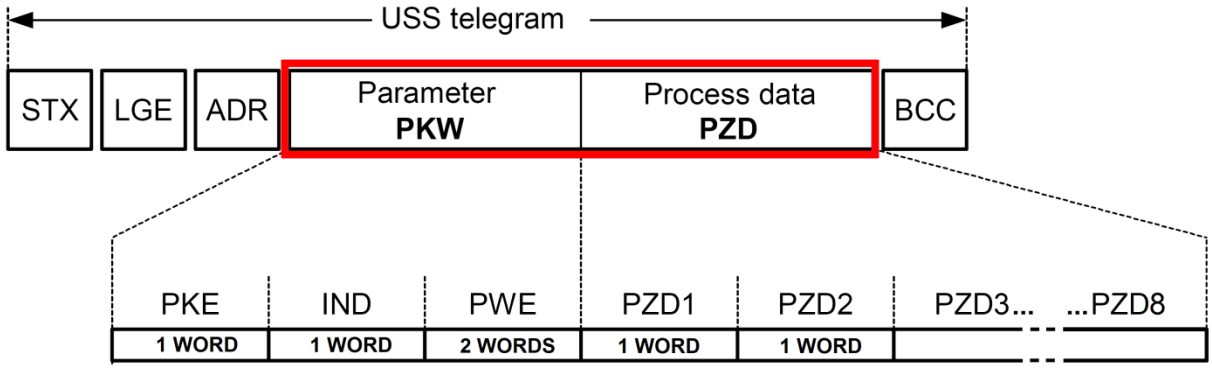


Figure 4.2 – Structure of use data block as we used it

4.2.7 Parameter channel PKW

The parameter channel can be used to monitor and/or change any parameter in the inverter. The inner structure is not trivial and can be looked up for details in the referenced literature [27], [28]. Simply, this block is used to enter the number of the parameter we want to operate (read or set), index as the basic parameters have different values for each Command (CDS) and Drive Data Set (DDS) [29], and the operation we want to perform (request or response identifier).

For acyclic communication via USS, the number of PWEs (third and fourth words) can vary. For 16-bit values, one PWE is required. If 32-bit values are exchanged, two PWEs are required. As we set the PKW to constant length of four words for our purposes, it means that the PWE area will always occupy two words. Then, a 32-bit parameter value comprises PWE1 (high-order word, third word) and PWE2 (low-order word, fourth word). A 16-bit parameter value is transferred in PWE2 (low-order word, fourth word). PWE1 (high-order word, third word) must be set to 0 in this case.

4.2.8 Process data channel PZD

In this area of the telegram, process data (PZD) are continually exchanged between the master and slaves. It contains the signals required for the automation. Dependent on the direction, the process data channel contains data for a request to the USS slaves or for a response to the USS master. In the requests are control words and setpoints for the slaves, and in the responses are status words and actual values for the master. The number of PZD words in a USS telegram is determined by parameter P2012. First two words are as follows:

- Request to USS slave: Control word 1(STW1) and main setpoint (HSW)
- Response to USS master: Status word 1 (ZSW1) and actual value (HIW)

If P2012 is greater or equal to 4, the additional control word (STW2) is transferred as the fourth PZD word (default setting) (Fig. 4.3).

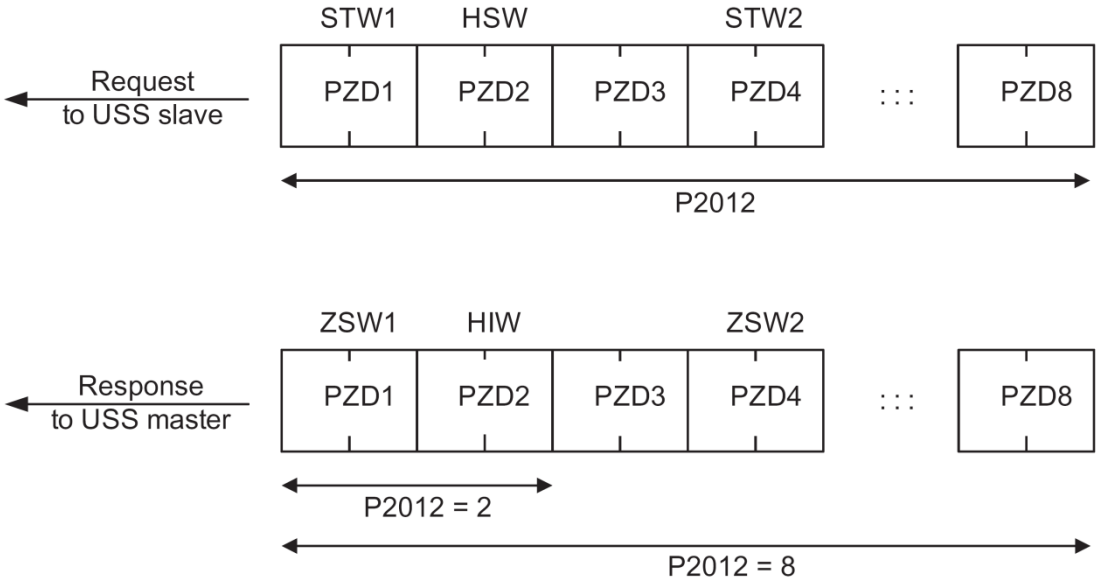


Figure 4.3 – Process data channel structure [28]

As we were also interested into continuous monitoring of torque and power of the electromotor, we set the PZD length to eight words, and by the mean of BICO technology, we linked some of the extra words so the channel structure was set as follows:

PZD1 = STW1/ZSW1
PZD2 = HSW/HIW
PZD3 = not linked
PZD4 = STW2/ZSW2
PZD5 = not linked
PZD6 = not linked
PZD7 = r0031 = Act. filtered torque
PZD8 = r0032 = Act. filtered power

Actual values of parameters such as setpoint, frequency, torque, and power are read through normalized values. The range can be from -200% (8000hex) to 199.99% (7FFFhex) of the reference value. It is therefore very important to set the reference values in expected range of the measurement to prevent exceeding. The range is coded as word (U16). It means that we get $400/2^{16}$ levels. That gives the resolution of 0.006% (Fig. 4.4).

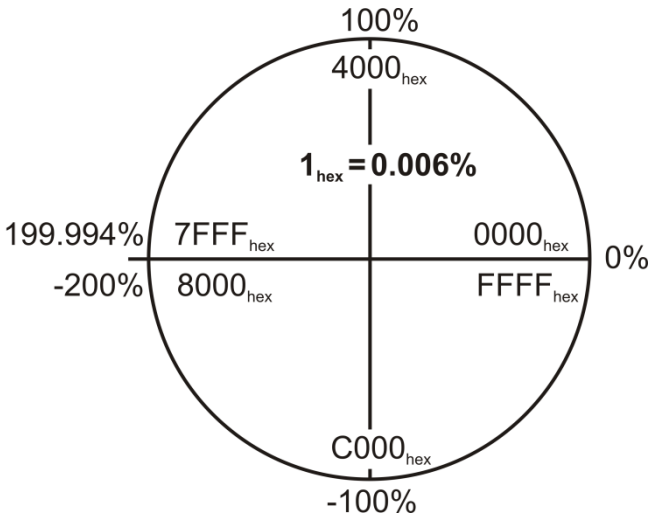


Figure 4.4 – Valid range and U16 coding of normalized values

4.3 Developed LabVIEW USS protocol library

As mentioned earlier, there was a constant need for creation of native LabVIEW API covering the most common commands and routines and enabling easy creation of new ones. We decided to develop LabVIEW USS protocol library that would serve for purposes of our measurements on the combustion engine. The final result was published for public use on National Instruments discussion forum – Motion Control and Motor Drives board. It is important to emphasize that the programmed library expects the length of the PKW to be set to four words (P2013) and PZD to eight words (P2012); otherwise, the communication will fail. As described in previous chapter, it also counts with PZD7 word to be linked to torque and PZD8 to the actual power of the electromotor.

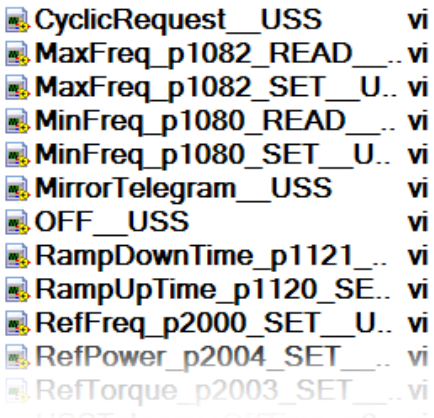


Figure 4.5 – List of created commands and routines (folder Commands)

The overall structure of the library respects the common rules of creating drivers or API in LabVIEW. The VIs (LabVIEW data file type) are divided into four main groups: Commands, HighLevel, LowLevel, and Examples. The group Commands (Fig. 4.5) contains our implementation of the basic commands that probably most of the users will take an advantage of. HighLevel and LowLevel folders consist of the support VIs that were used for the commands creation and can be used for the creation of new ones. Each of the commands

is equipped with an icon that from a quick glance offers the basic information about the command properties. If it is Read or Set-command can be determined from the color (Read = red, Set = blue, Special = cyan). Each of the commands is also equipped with short description information that helps to understand the aim of the command.

4.3.1 The command structure

We will describe the structure of the commands using one of the very basic ones – P1082, Maximal Frequency – Set. Other commands have structure, which more or less conforms to this one.

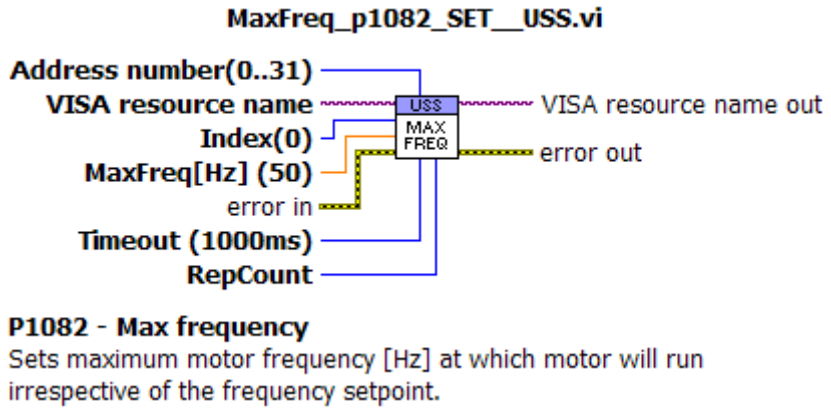


Fig. 4.6 – Connectors of single command and sample of description

Each of the encapsulated commands has connectors (Fig. 4.6) to enter the address of the inverter, VISA resource name that carries the information about the opened communication channel (baudrate, parity.), Index of the parameter we want to set (as mentioned earlier there can be different values for each CDS/DDS), and value to be set (read value in case of Read commands) showing also the default value. Error handling is present and thus if error occurs in any of the VIs, the following ones will be skipped and the error reported immediately. The remaining parameters are Timeout and RepCount, which are the parts of the resending algorithm for handling communication outages as described in the chapter *4.3.2 Communication outages on RS485*.

Inner command structure consists of five parts and can be seen in the Figure 4.7. The first element of the cluster marked (Fig. 4.7-1) sets whether the PKW part should be present in the telegram. If set to false, it automatically sends empty (=zeros) PKW channel. Next element is the request identifier specifying the desired action to perform. The parameter number, index, and value of the parameter follow.

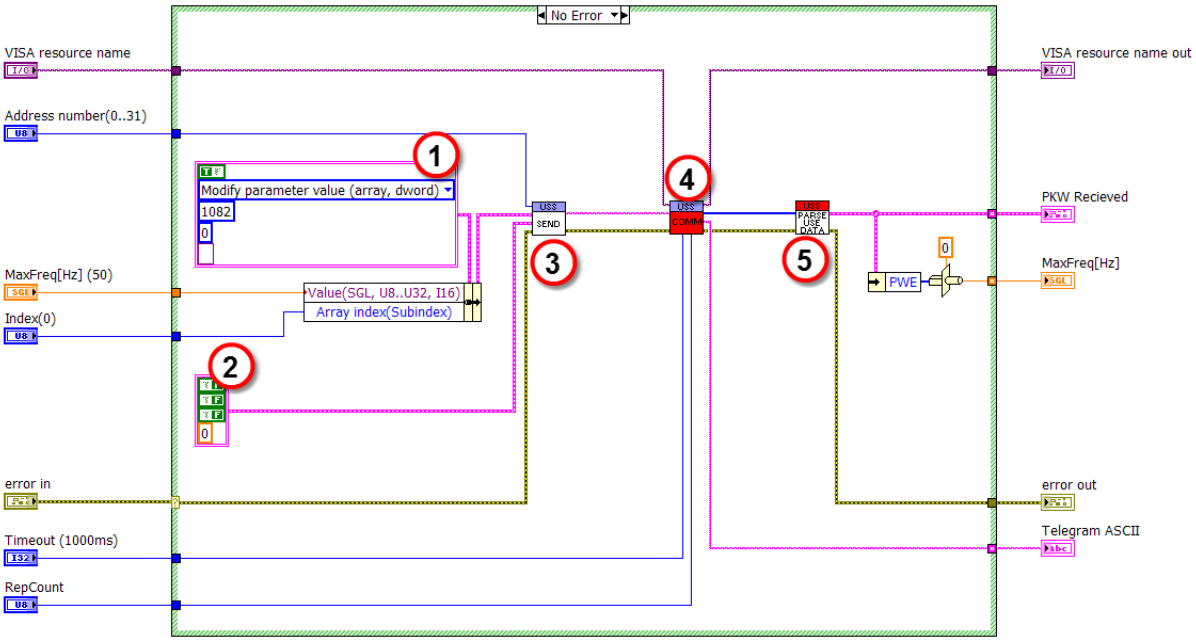


Fig. 4.7 – The inner structure of USS command

The first element of the cluster marked (Fig. 4.7-2) sets whether the PZD part will be used. Next element determines whether the inverter should be switched ON/OFF. Following elements are the inversion of the setpoint and the desired value of the setpoint itself (normalized value). Both mentioned clusters are type definitions created for easy manipulation and possible custom changes.

The VI Send_Telegram__USS.vi (Fig. 4.7-3) handles creation of the telegram to be sent according to the protocol specifications stated in the previous chapters and referenced literature [27]. The hierarchy of the VIs can be seen in the Figure 4.8. The purpose of each part can be determined from the SubVI name. The SubVIs can be found in the HighLevel or LowLevel folders depending how deep in the structure they are located. As a result, the exact telegram bit structure is composed and made ready to be sent to the inverter.

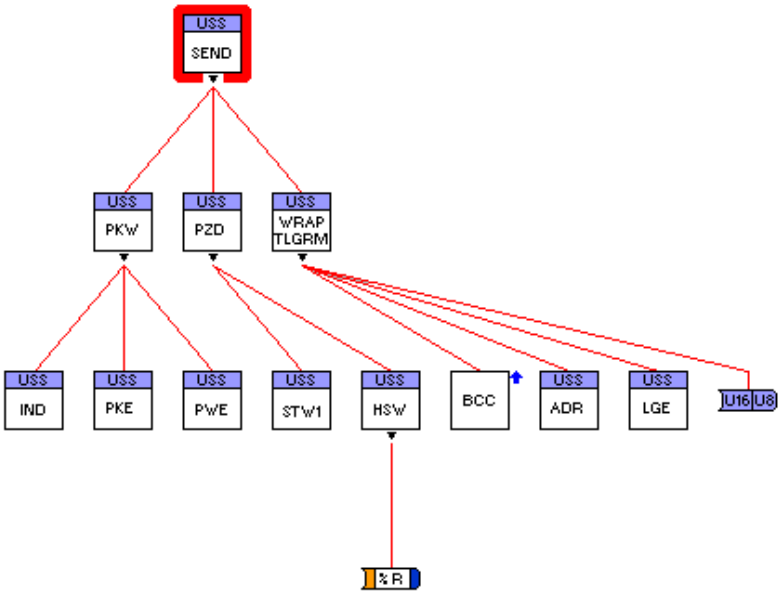


Figure 4.8 – Hierarchy of Send_Telegram__USS.vi

The next step (Fig. 4.7-4) is to send the telegram and receive the answer from the inverter. After sending, the port is continuously scanned. The received answer is parsed. BCC, the length and the presence of STX are checked. If the answer has unexpected format or is not received at all, the resending algorithm is applied or an error is reported. All mentioned is programmed in the Comm__USS.vi. In Figure 4.9, the block diagram and the VI hierarchy can be seen for better understanding. Respecting the data flow, the execution starts with the LabVIEW write primitive then continues to the Receive_Telegram.vi that continuously scans the port and as a next step the possible error checking is wired. The code is encapsulated with a while loop so that the actions can be repeated if there is a need to resend the telegram.

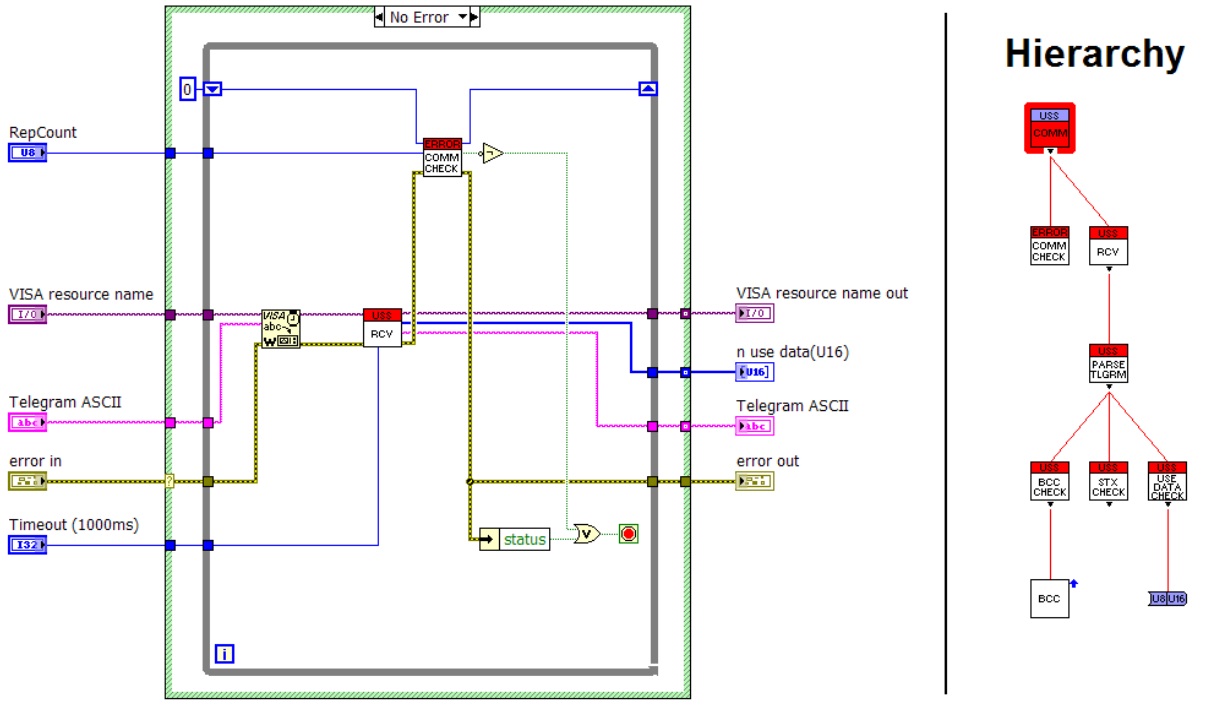


Figure 4.9 – Block diagram and hierarchy of Comm__USS.vi

The last part (Fig. 4.7-5) is focused on obtaining the important information from the received answer. In case of basic commands, it will be only the value we set/write and the inverter state, but in case of the setpoint adjustment we will also get the actual status words and the linked parameters (in our case frequency, power, and torque).

The created library is enclosed on CD.

4.3.2 Communication outages on RS485

Another problem that we came across was the random communication outages. The inverter time to time did not reply when a telegram was sent to it, most likely because it did not receive the request or the request arrived distorted, and thus it was refused. We used RS485 (EIA-485) to exchange the telegrams. RS485 is a two-wire, half-duplex, multipoint serial communication channel. As it uses a differential balanced line over twisted pair, it can span relatively large distances (up to 1200 m). Ideally, the two ends of the cable will have a termination resistor connected across the two wires. Without termination resistors, reflections of fast driver edges can cause multiple data edges that can cause data corruption. Termination resistors also reduce electrical noise sensitivity due to the lower impedance. Termination is

usually required for long distances and high speeds. The length of the cable we used was about 4 m and thus we did not terminate the line at first. Nevertheless, random outages in the communication appeared. We thought that the termination might help and we switched ON the termination DIP switch on the CU240S and connected the termination 220U resistor across the two signal wires on the opposite end of the cable (schema depicted in Fig. 4.10).

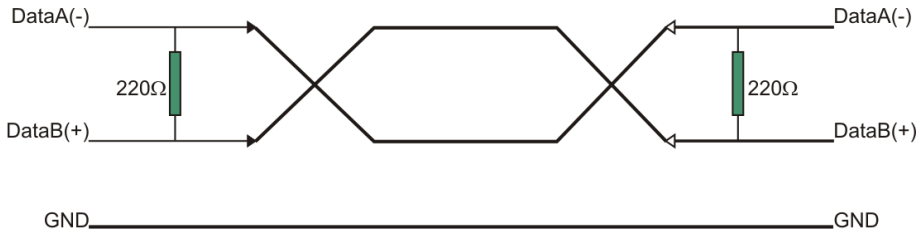


Figure 4.10 – Termination schema

We also connected the bias resistors, but none of those adjustments helped to prevent the communication outages. As those problems appeared randomly but rarely, we decided to implement simple resending algorithm (Fig. 4.11) to the API. As it was described in the chapter *4.3.1 The command structure*, each command VI is equipped with connectors Timeout and RepCount. Timeout defines the time during which the program has to receive the valid response. If the valid response does not arrive, it will resend the data in predefined number of attempts. If none of those attempts succeeds, the error is reported indicating that there is a major problem in the communication.

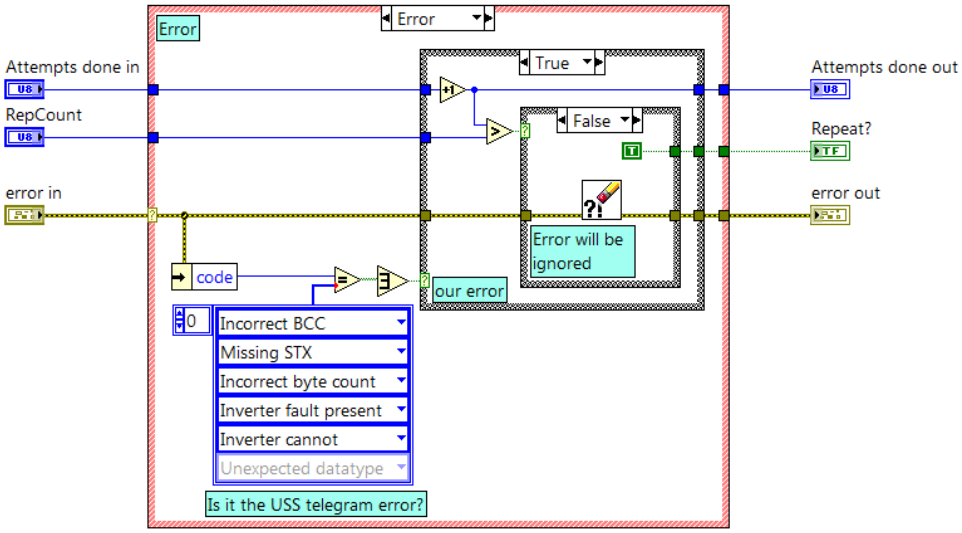


Figure 4.11 – The core of the resending algorithm

4.3.3 Slave response latency

During the period of testing programmed USS library, we also encountered unexpected behavior of the response telegrams send by the inverter. Communication with Siemens inverters is focused on cyclic communication. As the typical scenario is for the master (typically a PLC) to continuously scan multiple slave devices, command data refresh rate is more important than response data latency. Rather than make the master wait for the slave to

formulate a reply on every pass, the slave has the reply ready for the master based on the requested data in the previous pass. This way communication time is kept to a minimum. This behavior unfortunately might complicate the acyclic communication when we just want to set some parameter of the inverter. The parameter will be changed correctly (if no fault or error occurs), but the immediate response of the slave will consist of the value of the previously set/requested parameter. The same and even more confusing situation occurs when requesting only value of some parameter. Instead of sending the reply to requested parameter, it sends reply to the previous request. For example, if we asked the value of parameter P1082 (maximum motor frequency) sending following telegram (PKW = 4, PZD = 8):

```
02 1A 00 64 3A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 46hex
```

64 $3A_{hex}$ = P1082

we would receive the answer with the value of parameter P1080 (minimum motor frequency) that we requested before:

```
02 1A 00 54 38 00 00 00 00 00 00 EB 31 00 00 00 00 06 E0 00 00 00 00 00 00 00 00 48hex
```

54 38_{hex} = P1080

This behavior is necessary to take into account when dealing with any implementation of the USS protocol. If you parse the entire reply, you always know to which parameter you apply the data, but it won't probably be the same you set/asked.

In version 1.2 of the library, support for obtaining the value of the actual requested command was added. Basically, the API queries the controller until it receives the proper answer (Fig 4.12).

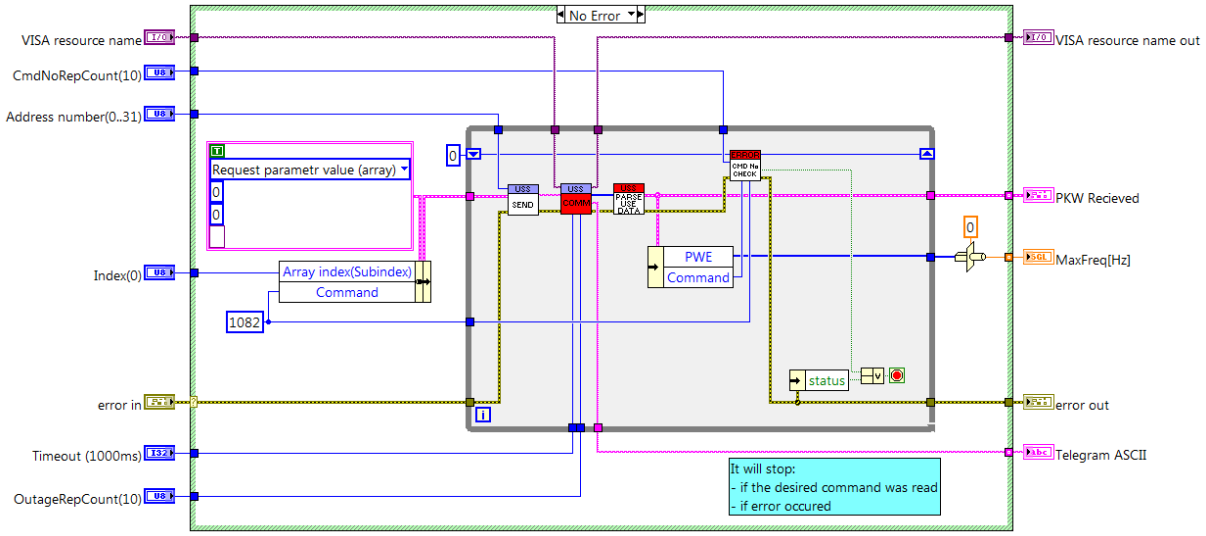


Figure 4.12 – Repetitious querying of command value

4.4 USS protocol implementation summary

We created API for handling the communication with the family of SINAMICS inverters via the USS protocol from superior PC-based system. The library was tested with the G120 inverter and control unit CU240S, and it proved its reliability. As it was given for public use, other users already reported its usability with other types although some minor changes had to be made in some cases. The creation of the library was motivated by the nonexistence of any implementation of the mentioned protocol into LabVIEW, which is nowadays probably the most spread environment for carrying out custom measurements and PC-based automation. The library is also supplemented with few examples, where one presents simple application for driving the electromotor and setting the very basic parameters programmatically.

5 Frequency measurements

This chapter describes three methods for making frequency measurements (to determine the shaft rotational speed and the shaft speed fluctuations) with the counters/timers of the PCI 6602 board, and explains how to determine accuracy. It also selects the most suitable method for our purposes.

To perform frequency/speed readings by means of the IRC TTL pulses, the counter card is equipped with NI-TIO, a sophisticated counter and digital I/O ASIC with eight 32-bit counters [30].

5.1 Frequency measurement error

There are two sources of error in frequency measurements. The total measurement error is the sum of these two sources [31]:

- Errors in the frequency of the crystal oscillator
- Measurement error, e.g. the measurement uncertainty

The crystal oscillator used on the PCI-6602 is rated at 50 parts per million (ppm) stability over temperature [30]. 50 ppm means that for every 1 million Hertz, the frequency can be off by ± 50 Hz. For a 6602 board, which has a 80 MHz oscillator, the error due to crystal used could be up to:

$$\frac{80\ MHz}{1\ MHz} \cdot \pm 50\ Hz = \pm 4000\ Hz \tag{5.1}$$

As a result, the timebase is not going to make a significant difference in our measurement (compared to the measurement uncertainty, e.g. measurement error). We will go through some numbers to show that. Let us say we have the following situation:

- We have IRC mounted at the end of the camshaft
- The IRC ppr is set to 3600
- The actual $rpm_{crank} = 4500 \rightarrow f_{cam} = 37.5$ Hz
- The f_k (known source frequency) = 80 MHz (timebase is used)
- We use the single counter method as described in chapter *5.1.1 Method 1 – Inverse period measurement*

The pulses would then be coming in at f_p = 37.5 x 3600 = 135 kHz. During each pulse, we would expect to count 592 or 593 ticks of the 80 MHz timebase depending on the relative phase between the timebase and the encoder signal. If the timebase was 80,004,000 Hz or 79,996,000 Hz instead, based on the calculation (5.1), we would still receive 592 or 593 ticks in this case. The frequency of the IRC TTL pulses would have to be orders of magnitude higher in order to generate a whole extra pulse during the frequency measurement. So we can neglect the crystal oscillator error for our measurements and fully focus on the measurement errors.

The measurement errors are inevitable, but can be minimized by choosing a frequency measurement method that is most suited for the frequencies measured and the timebase of the device used. There are three different methods that can be used for the frequency readings that

are discussed in next chapters. The Matlab code used for necessary calculation can be found in *Appendix 1 – Frequency measurement error calculations.*

5.1.1 Method 1 – Inverse period measurement

This is the simplest method that needs only one counter and is suitable for low frequency measurements. The period is measured by counting the rising or falling edges of a known source frequency between the two consecutive rising or falling edges of the unknown frequency. By taking the frequency of the known source and dividing by the count, we can calculate the period of the unknown signal. The internal 80 MHz timebase of the counter card is used as the source of the known frequency (Fig. 5.1).

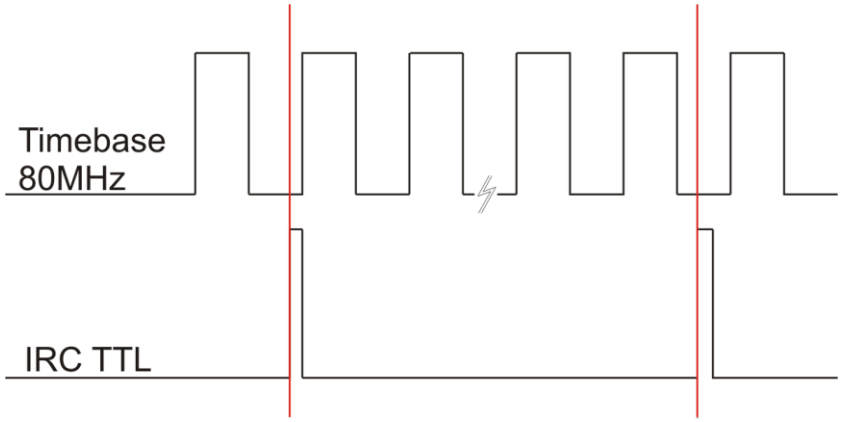


Figure 5.1 – Inverse period measurement

We will explain the measurement error calculation for the situation when:

- We have IRC mounted at the end of the camshaft
- The IRC ppr is set to 720
- The actual $rpm_{crank} = 4500 \rightarrow f_{cam} = 37.5$ Hz
- The $f_k = 80$ MHz (timebase is used)

The resulting frequency of the pulses detected by the IRC sensor at the given speed is then $f_p = 27$ kHz. Assuming that the 80 MHz timebase is perfect, in other words the crystal oscillator used to generate the 80 MHz timebase is perfect, we should count 2962 edges between two consecutive rising or falling edges of the perfect 33 kHz signal. However, due to the fact that we cannot control the phase relationship between the two signals (Fig. 5.1), the count can be 2962 or 2963. This is called the measurement error. The detected rpm of the crankshaft is given by:

$$f_p = \frac{f_k}{count}\ [Hz] \tag{5.2}$$

$$rpm_{crank} = \frac{f_p}{ppr} \cdot 60 \cdot 2\ [Hz] \tag{5.3}$$

Doing the math gives the following results:

Table 5.1 – Calculated rpm_{crank}, IRC ppr =720; rpm_{crank} = 4500

Count returned	2962	2963
Calculated rpm_{crank}	4501	4500

The deviation from the actual frequency is acceptable. It is obvious that if the frequency of the pulses is higher, the error will increase. The same situation with IRC ppr set to 3600 (f_p = 135 kHz) gives the following results:

Table 5.2 – Calculated rpm_{crank}, IRC ppr =3600; rpm_{crank} = 4500

Count returned	592	593
Calculated rpm_{crank}	4505	4497

You can see that a difference of one count results in a larger sampling error as the frequency to be measured increases. This method works well as long as the frequency of the signal to be measured is significantly slower than the known source frequency. As the frequency of the signal to be measured increases to approach the frequency of the source frequency, measurement error increases.

To calculate maximal measurement error using this frequency measurement method, we have the following formulas:

$$Err = f_x \cdot \frac{f_x}{f_k - f_x} \; [Hz] \tag{5.4}$$

$$Err = \frac{f_x}{f_k - f_x} \; [\%] \tag{5.5}$$

where f_x is the unknown measured frequency (in our case f_p) and f_k is the known frequency (in our case the 80 MHz timebase). For the first situation (IRC ppr = 720; rpm_{crank} = 4500) it gives the maximal possible error of 0.03% = 2 rpm. For the second situation (IRC ppr = 3600; rpm_{crank} = 4500) the maximal possible error is 0.17% = 8 rpm.

5.1.2 Method 2 – Count number of pulses in known time

In this configuration two counters are used. One counter will count the number of edges of the unknown high frequency during a period of the known signal (generated by the other counter) (Fig. 5.2). This approach is suitable for higher frequencies. The larger the period of the known signal used to gate the counting, the smaller the measurement error.

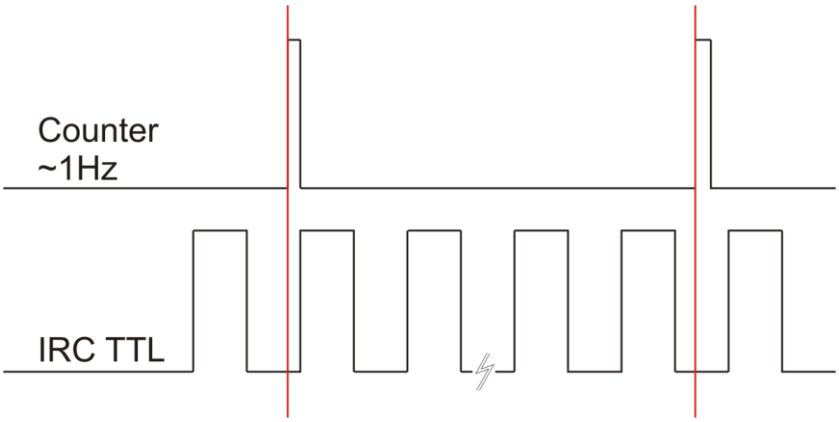


Figure 5.2 – Count number of pulses in known time

With this method we could achieve very good accuracy if we used the frequency of the second counter around 1 Hz. By using the equations (5.6) and (5.3) we can calculate the resulting rpm_{crank}. From the Table 5.3 we can see that we would always count accurately enough.

$$f_p = \frac{count}{T_{counter}} \; [Hz] \tag{5.6}$$

Table 5.3 – Calculated rpm_{crank}, IRC ppr = 720; rpm_{crank} = 4500

Count returned	26999	27000	27001
Calculated rpm_{crank}	4500	4500	4500

Nevertheless, this method is not suitable for our measurement, since we want to obtain as many frequency readings as we have ppr. To achieve good accuracy we would have to use very low frequency of the gating counter ~ 1 Hz which would result in only one reading per second. As a result, we would have no information about the speed fluctuations during one revolution of the shaft.

5.1.3 Method 3 – Measure time of known number of cycles

The method which was mentioned first is suitable for measuring low frequencies. The second method can be used for measuring high frequencies. However, if we need to span a wide frequency range covering both low and high frequencies, then this third measurement method can be considered.

This measurement method also uses two counters. The first counter is used to divide down the frequency of the signal to be measured and the second counter is used to measure the period of the divide down frequency (Fig. 5.3). The actual frequency can be calculated by multiplying the resulting frequency measured by the divide down value *N* (Eq. 5.7).

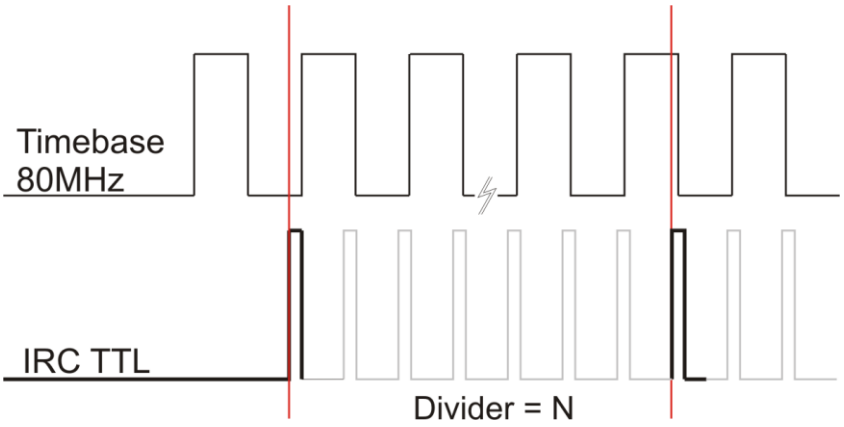


Figure 5.3 – Measure time of known number of cycles, divider = 7

$$f_p = \frac{f_k}{count} \cdot N\ [Hz] \tag{5.7}$$

Table 5.4 – Calculated rpm_{crank}, IRC ppr = 3600; rpm_{crank} = 4500; divider N = 4

Count returned	2370	2371
Calculated rpm_{crank}	4501	4499

With this frequency method, the larger the divider value, the slower the resulting frequency, and the more accurate the measurement result. The measurement error is calculated as for Method 1 but with respect to the divider *N*.

$$Err = f_x \cdot \frac{f_x}{(N \cdot f_k) - f_x}\ [Hz] \tag{5.8}$$

$$Err = \frac{f_x}{(N \cdot f_k) - f_x}\ [\%] \tag{5.9}$$

For our example the worst possible error is then 0.04% = 2 rpm. As an inevitable side effect of this method, the amount of rpm readings is reduced, e.g. if we set the IRC to provide 3600 TTL pulses and the divider *N* = 4 then we end up with 900 rpm samples.

5.1.4 Which method to use?

From the above text, it is apparent that Method 2 is not suitable for our measurement type and the frequency range we expect to measure. Since we demanded the same amount of rpm readings as we have TTL pulses available we initially chose Method 1. This method works perfectly for lower frequencies and since we mostly use the IRC with ppr set to 720 to achieve the 0.5° resolution of the cam, we also reach satisfactory accuracy.

The problems arise when we switch the IRC to 3600 ppr. After some time of rpm reading (the time depends on the shaft speed) we receive an error that the data was overwritten before it could have been read by the system. After a short investigation we found out that this error refers to samples being overwritten on the hardware buffer (i.e. onboard FIFO). The PCI-6602 is not the newest counter board and the on-board FIFO is only 2 samples [30]. If the pulses are latching in values faster than we can pull them off of the board (to the DMA buffer) this error occurs. A similar chip was benchmarked [32] for continuous

counter measurements at 150 kHz, although this number is known to be highly system dependent.

We conducted a benchmark for our system in which we identified the maximal frequency of the incoming TTL pulses that the counters of the card were capable of measuring. We used a signal generator. The results are compared with other available DAQ cards equipped with programmable counters – see Table 5.5. Stated frequencies are the maximal frequencies that would not cause a buffer overflow during continuous measurement over an extended period of time. Our system specifications are listed in *Appendix 2 – System parameters.*

Table 5.5 – Benchmark results

NI PCI 6221	135 kHz
NI PCI 6120	85 kHz
NI PCI 6602	135 kHz

The NI PCI 6221 and NI PCI 6602 have the same NI-TIO counters which as expected results in the same maximal frequency. On the other hand the NI PCI 6120 has the prior DAQ-STC chip which has a lower maximal value.

The NI PCI 6602 was successfully able to measure the pulse train up to 135 kHz. It is slightly less than the earlier mentioned 150 kHz. It was said that this number is highly system-dependent. The main reason for the lower value is that our card was placed in an expansion chassis and connected via ExpressCard module.

For a 3600 ppr, 135 kHz would give a maximum rpm_{crank} = 4500 if the IRC was mounted at the end of the camshaft. Since we conduct the kinematics measurement usually up to 6000 rpm (in the case of racing valvetrains even up to 8500 rpm) we decided that for cases when the IRC ppr is switched to 3600 ppr, Method 3 will be used. Based on the principle of Method 3 we won't encounter the above mentioned error.

As a result we dynamically switch Methods 1 and 3. Method 3 with $N = 4$ is used if the frequency of the pulses is potentially higher than 100 kHz, e.g. if IRC is set to 3600 ppr. Thus, we keep an acceptable amount of speed samples and good accuracy, and still have 3600 pulses available for analog data acquisition. With the IRC ppr set to 720 we still use Method 1.

5.1.5 Camshaft speed fluctuations and the IRC accuracy

We chose two frequency measurement methods suitable for our task. Method 1 (M1) offers the accuracy of 0.03% for ppr = 720 and Method 3 (M3) of 0.04% for ppr = 3600 and divider $N = 4$ at $rpm_{crank} = 4500$. We would achieve this accuracy if we had a perfect and constant signal to measure, e.g. from a signal generator.

We performed three independent measurements of camshaft speed fluctuation that were run sequentially with different settings. After data smoothing, we clearly see the expected speed fluctuations which are given especially by the valve spring forces, the shape of the cams themselves and chain vibrations [33]. Besides that, we encountered noise both with Methods 1 and 3 (Fig 5.4).

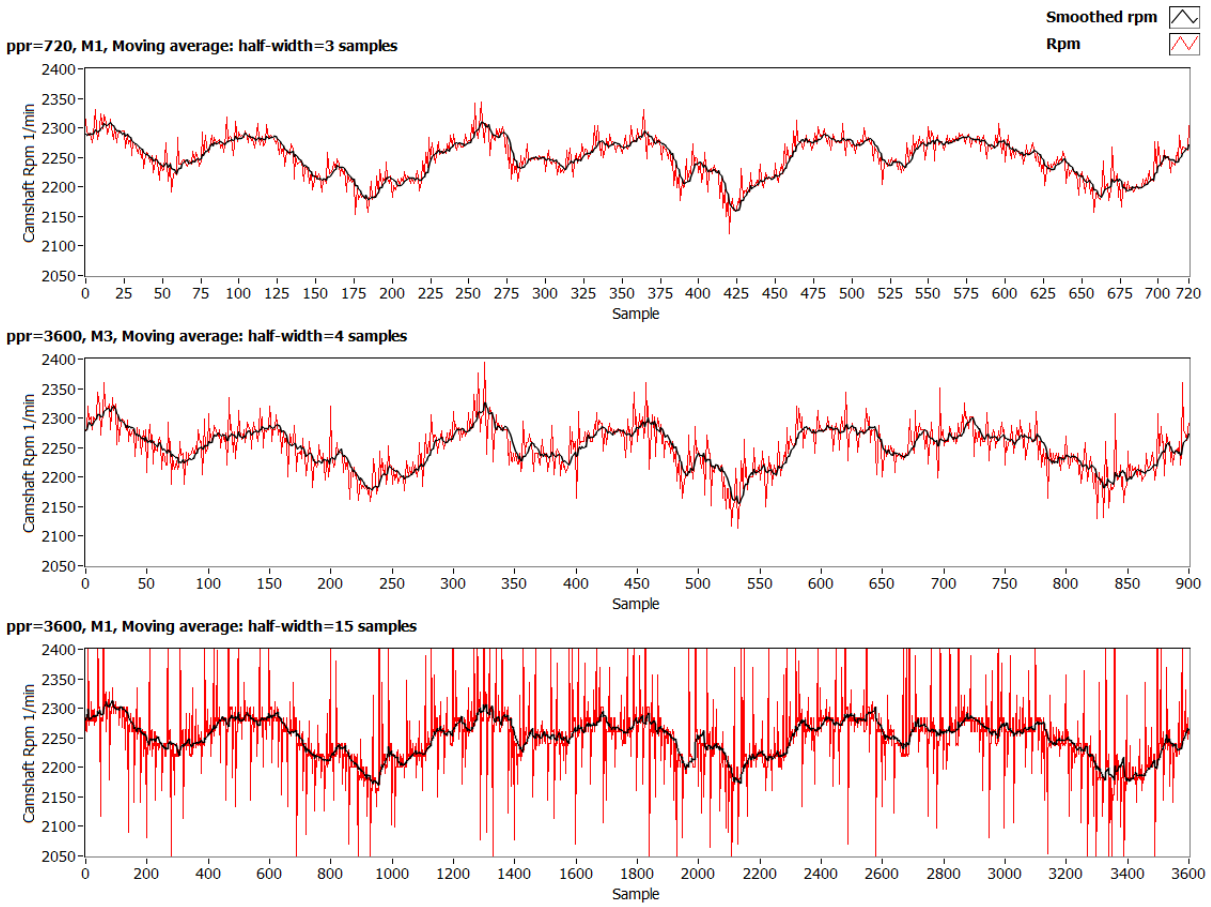


Figure 5.4 – Camshaft speed fluctuations, rpm_{crank} = 4500, intake camshaft, 3-cylinder engine, comparison of different frequency measurement methods and ppr

The noise is the variance in the output of the IRC. Basically, since the IRC is a mechanical device the pulses would not be 100% constant even at constant speed, thus allowing extra pulses of the 80 MHz timebase to be counted or missed. Another source of error with ppr set to 3600 is the multiplier connected to the IRC. The IRC itself has 720 segments and higher resolutions are achieved with the multiplier which introduces additional uncertainty. Method 3 essentially averages multiple consecutive period measurements together before inverting to obtain frequency, which explains the lower amplitude of the noise. The moving average filter was used to smooth the data.

As can be seen in Figure 5.5, all the methods with proper postprocessing deliver the same results. Slight differences are expected since the results were obtained from three consecutive measurements.

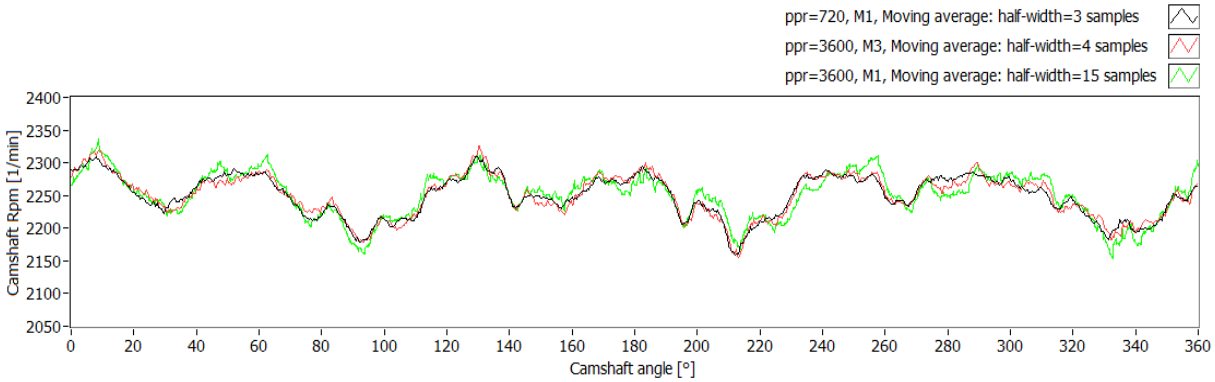


Figure 5.5 – Camshaft speed fluctuations, rpm_{crank} = 4500, intake camshaft, 3-cylinder engine, comparison of different methods with different ppr and different postprocessing

The parameter *Dynamic accuracy at 10,000 1/min (signal delay)* in Table 3.2 quantifies the accuracy of the IRC sensor. If the shaft revolves at 10,000 rpm the angle between any two consecutive pulses can be 0.02° wider or narrower than the expected resolution. Let us consider this extreme case while using Method 1 and having ppr set to 3600. It means that in the worst case the period of two consecutive pulses can be 20% off. As a result, it could produce a deviation of 2000 rpm from the real value. It is important to emphasize that it still applies to the error of a single speed measurement between two consecutive pulses. In this situation, the frequency measurement error is neglectable in comparison with IRC accuracy. We can say the same about the lower IRC ppr. For 720 ppr the expected resolution is 0.5°. Since it can be ±0.02° off, the IRC speed reading error is ±4%.

Recapitulation of different methods, resolutions and corresponding measurement uncertainty and IRC accuracy can be found in Table 5.6. The exact dependency of the rpm and the accuracy of the IRC sensor is not known, although the accuracy is expected to be slightly higher with lower speeds.

Table 5.6 – Comparison of the frequency errors and the IRC accuracy

rpm_{crank} = 4500	Errors in the freq. of the crystal oscillator [%]	Frequency measurement error, i.e. measurement uncertainty [%]	IRC error [%] (at 10,000 rpm)	Recommended half-width of the Moving average filter [points]
ppr = 720; M1	neglectable	0.03	4	3
ppr = 3600, M3, divider = 4	neglectable	0.04	20/4	4
ppr = 3600, M1	neglectable	0.17	20	15

To see if the IRC sensor has the expected accuracy, the smoothed signals from Figure 5.4 were taken as reference signals. For each measured speed signal the deviations from its reference were calculated and converted to percentage as

$$deviation\ [\%] = \frac{rpm_{cam\ ref} - rpm_{cam\ meas}}{\frac{rpm_{cam\ ref}}{100}} \tag{5.10}$$

where $rpm_{cam\,ref}$ is the smoothed signal, $rpm_{cam\,meas}$ is the unchanged measured signal of the camshaft speed fluctuations and $\overline{rpm}_{cam\,ref}$ is the average of the reference signal. Figure 5.6 depicts the results. If we compare Table 5.6 with the picture, we see that all the values are in the expected range. Only two values in the figure depicting the M1 with ppr = 3600 exceeded the 20% range. Considering that our reference was obtained by smoothing the measured data we can consider this as a very good result which verifies the accuracy of the IRC sensor.

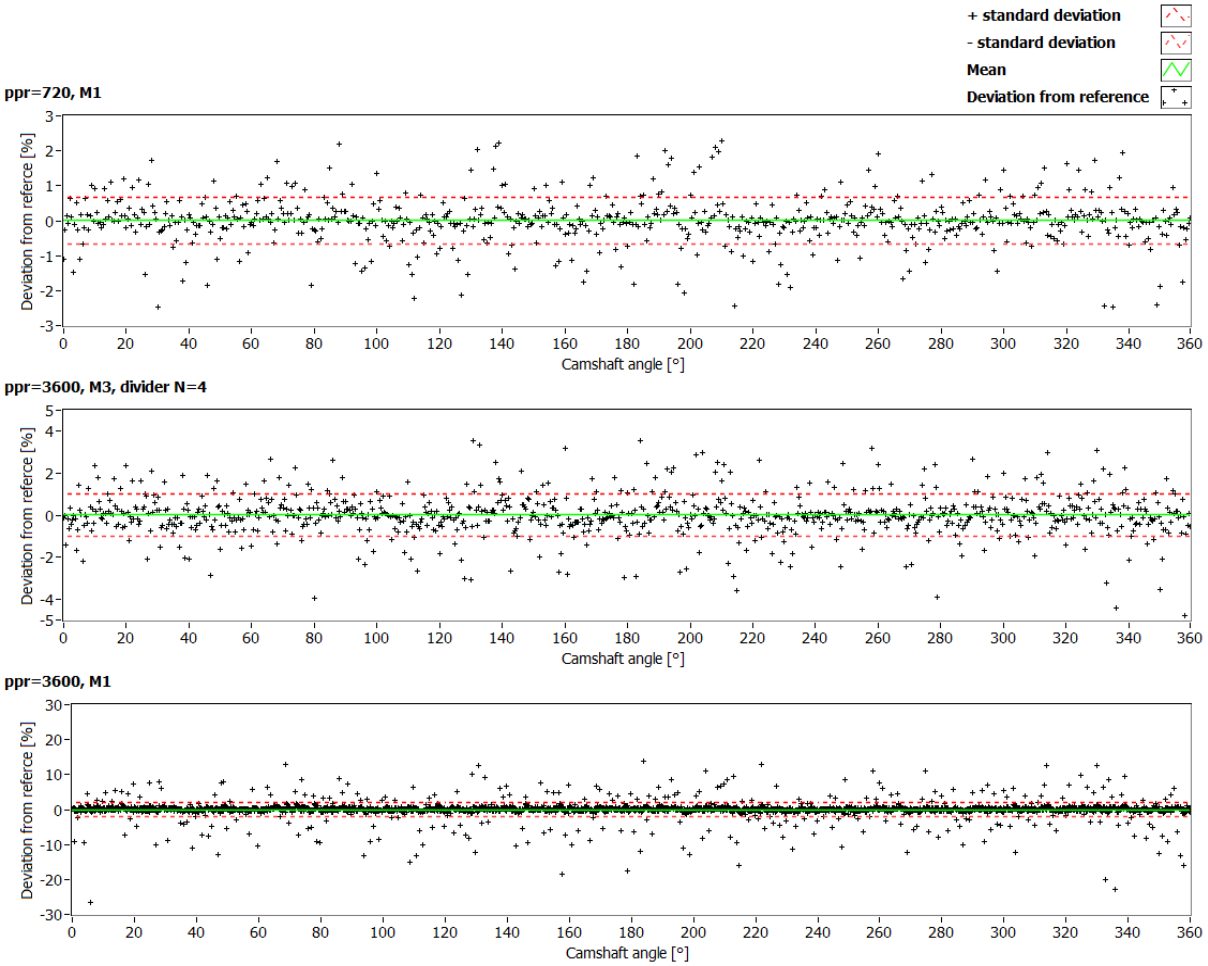


Figure 5.6 – Accuracy of the IRC sensor, rpm_{crank} = 4500, intake camshaft, 3-cylinder engine, comparison of different frequency measurement methods and ppr

6 Additional counter tasks

Since we use a counter card with 8 counters and for the frequency measurements we utilize only up to 2 counters, we have plenty of counters left for other tasks.

6.1 Automatic ppr identification

We often switch between different resolutions of the IRC ranging from 720 to 3600 pulses per revolution. To represent the speed readings correctly, and for additional calculations, it is necessary to know this value. For this reason we implemented a routine (Fig. 6.1) that is run once the measurement starts and identifies the ppr. This way it is not necessary to manually enter the value every time we change it. It employs a simple buffered edge counting using the period measurement (the same principle as Method 1) which will return the period in ticks of the timebase as the units. The timebase in this case is the IRC signal itself and the gating signal is the IRC trigger mark. We have to dispose of the first read value since it is not an accurate representation. It poses a measure of the number of edges of the source signal between the beginning of the measurement and the first rising edge of the gate.

This routine tends to deliver false readings during the first few revolutions of the electromotor. The electromotor produces a strong magnetic field during the peak starting torque which propagates to the trigger signal and produces false trigger marks. Thus it is advised to utilize the identification routine after the first few seconds of revolution, at which point the starting torque drops.

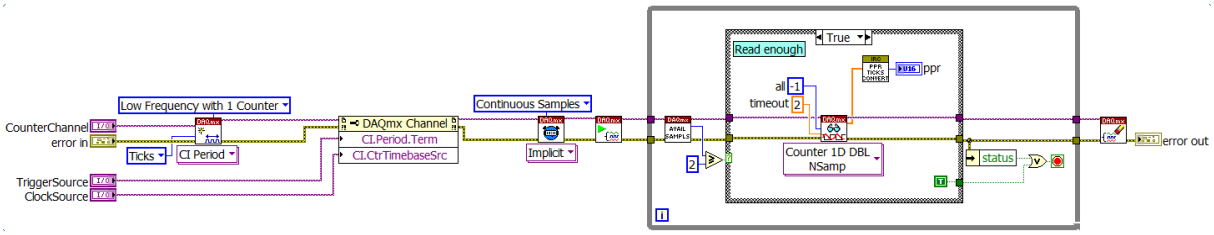


Figure 6.1 – The ppr identification routine

6.2 Trigger pulse delay

Another desired operation is to delay the IRC reference pulse that is generated at each rotation and denotes the shaft zero position, triggers our measurements (valve kinematics and shaft speed fluctuations) and resets the laser probes controller. The controller has to be reset at every revolution, since the displacement measurement is more sensitive to DC movements, thermal effects and environmental influences.

When we acquire the data we always want to have a record that starts on the base circle, covers the valve opening and closing phases and finishes on the cam base circle again covering the whole 360 degrees. Since the reference mark is fixed, it can happen that after the mounting of the sensor at the end of the camshaft the pulse would be generated during the opening/closing phase of the valve. That would start the acquisition at an unwanted position and also resetting the probes at this point would cause false reading. Traditionally, the IRC

would have to be remounted. We could just delay the start of our measurement for a predefined amount of IRC pulses but we also need to generate a corresponding pulse to reset the laser probes at the new trigger position.

The solution is to utilize one counter for pulse generation. When the original trigger (reference pulse) arrives, it generates a new pulse with predefined delay. The delay is specified as the amount of ticks of the source signal which is the IRC TTL pulse train. The delay can be automatically determined by an algorithm which assures that the measurement will always start in the same position (e.g. the maximal valve displacement will always be located at 180°). The newly generated pulse serves as a new trigger for resetting the probes and starting the measurements. The operation is set as retriggerable to perform on every recognized trigger (Fig. 6.2).

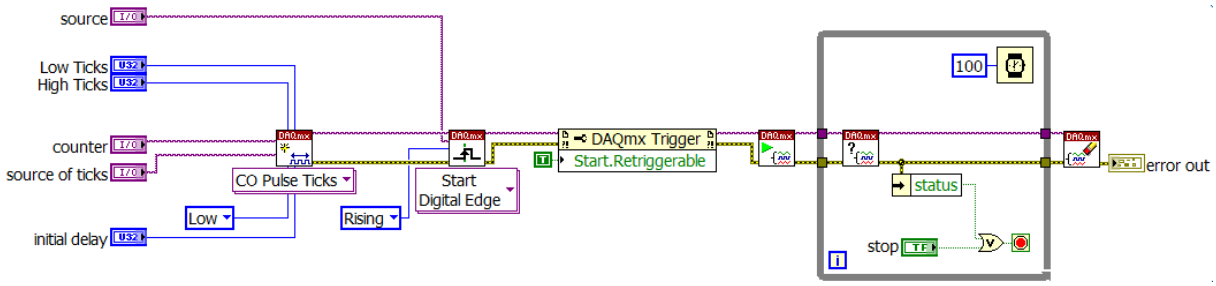


Figure 6.2 – Reference pulse delaying technique

7 Architecture patterns used for the software creation

This part of the work focuses on the architecture patterns that were necessary to combine to assure such a complex task as fully automated measurement of kinematic variables of valvetrains and, in parallel, real-time shaft speed fluctuations measurement.

7.1 State Machine

Each technical application can be described with more or less complicated state diagram. The architecture of State Machine (SM) can be used to implement complex decision making algorithms represented by state diagrams or flowcharts [34]. Each state performs something different and specific, and after applying the decision-making logic, it unambiguously leads to one or more states. The output function is thus determined by the current state alone. This matches the Moore machine.

The state approach is one of the advanced and often-used programming techniques in LabVIEW. Besides its ability to implement decision-making algorithms, SMs offer efficient planning. They are relatively easy to create and rather easy to modify just by adding or removing particular states or by changing their functionality.

To represent the state diagram in LabVIEW, we need at least the following programming structures:

- While loop – ensures the transition to other state
- Case structure – each case contains a code which is executed for a given state
- Shift register – transfers information about next/current state
- Decision-making logic – determines the next state

7.2 Queued State Machine

SM architecture is universal; nevertheless, it has its limits. If the volume of a project expands, it is not easy to force the SM to perform all the requisite operations. In such a situation, it is efficient to resort to the so-called *Queued State Machine* (QSM).

Notation for describing the characteristics of a queuing model was first suggested by David G. Kendall in 1953. Kendall's notation introduced an A/B/C queuing notation that can be found in all modern standard works on queuing theory, for example [35].

QSM is a LabVIEW programming pattern [36] that sends commands and other data from multiple source points (i.e., producer points), such as from user events and from one or more parallel processes, and gets these handled in one SM process (i.e., destination consumer point) in the order in which they were added to the queue [37].

An example of application can be multiple parallel programming such as parallel DAQ, communication with several instruments, monitoring, postprocessing where this method empowers any parallel VI to send and receive commands and data across other parallel VIs with no data loss.

7.3 Producer-Consumer design

During the measurement process where many time-critical operations (e.g., instruments communication) are performed, the data are acquired and analyzed at the same time; problems with data-processing rate may appear.

During one revolution of the camshaft, we measure at 2-4 channels (valve displacement and velocity, oil temperature and pressure or other variables such as spring force) 720 to 3600 elements of double precision type (DBL). The speed of the camshaft is acquired at the same time (1D array of the same size). The incoming data undergo further processing. The acceleration is computed, order analyses and other operations are performed; data are written into the hard-drive and displayed in charts/graphs. The whole time we also monitor crucial values of the electromotor (speed, torque, power) that also have to be saved and displayed. As a result we have multiple loops running at different rates.

With rpm_{crank} = 5000 we acquire a new data set each 24 ms (IRC mounted at the end of the camshaft). Described operations are, in general, time-consuming and the speed of processing highly depends on the CPU/system performance and load. So, it is necessary to find such an approach where the data reading is not dependent (doesn't have to wait) on its processing. We want to read the data from the card buffers as fast as we can to prevent their overflow and to have actual values of critical parameters. We also need to synchronize the data write calls to keep the data streaming on an efficient level. On the other hand, in order to keep the resources at a minimum, and also since the human eye has its limitations, we don't need to display the results that often.

These demands lead to the so-called Producer-Consumer design. It is based on a Master-Slave pattern and is geared towards enhanced data sharing between multiple loops running at different rates. In our case, the loops that acquire data (Producers – ELM, IRC, DAQ loop; see chapter *8 DREAM application*) run faster and independent of the loop that processes the data (Consumer – Processing loop).

7.4 Action Engine

Action Engines (AEs) behave as very efficient global variables. They can be thought of as machines (engines) that perform a useful task (action) often on something; typically but not strictly, data in Un-initialized Shift Registers (USRs). For detail see [38].

We use AE to store and reach data from the IRC readings and to perform some statistical computations on them. AEs can only be executing in one context at a time but can be called from different parallel loops. LabVIEW will ensure that if there is ever a situation where two contexts are attempting to act on the same AE at the same time, LabVIEW will schedule the first and the subsequent will wait until the first call completes. This behavior is employed to eliminate race conditions [38].

8 DREAM application

The apparatus for measurements of the valvetrain dynamics is very complex and the software covering it must be versatile and capable of performing tasks in parallel. It is widely recognized that multithreaded architecture poses significant programming challenges. LabVIEW offers an ideal programming environment for such a task because LabVIEW applications are inherently multithreaded [39]. We decided to use the techniques mentioned in previous sections to perform the measurements. QSMs with Producer-Consumer architecture are capable of sharing data and states between loops running in parallel. They enable parallelism of the code, and the queue itself can handle simultaneous reads and writes. In addition, queues are memory efficient – they do not create any copies of the data on their own.

The synchronization of the processes when dealing with multiple parallel loops is a significant issue. Most of the time the loops should run truly parallel, but to accomplish certain tasks it is necessary to synchronize them and assure determinism and sequentiality. We developed a solution that adds this capability by using the *Notifiers* [40]. When the states are queued, a notifier is created and can be tracked to inform the application whether or not the state has already been processed. Queuing of other states of any QSM can then be halted until the desired states are processed.

The crucial operations were to assure safe driving of the electromotor connected to the combustion engine, and in addition to read the electromotor values. At the same time to perform DAQ and measure the speed fluctuations of the camshaft; display, process and save all the data. This means that processing of the data set from one revolution has to be done while new data are being acquired to prevent buffer overflow. The benefit of the queue is obvious here. In a situation where the processing of the data takes longer than the acquisition of the new data set, the application doesn't have to wait until the processing is done but performs it all in parallel. New acquired data are added to the end of the data queue and are processed as soon as the loop finishes processing the previous data set. This approach saves time and, more importantly, data cannot be lost.

The developed application consists of five QSMs and was named DREAM. DREAM stands for Dynamic REsponse Automated Measurements. Figure 7.1 represents the application data flow diagram. Each QSM runs at its own rate. The solid blue lines represent communication between the QSMs where the elements are states together with variant data type (to exchange optional parameters). The dashed orange lines represent data queues meant for transferring the data to the processing loop. A description of the exact purpose of each of the loops is outlined in the following sections. Application state charts and printscreens are enclosed in the *Appendix 3 - DREAM application state charts and printscreens.* The source code of the application can be found on enclosed CD.

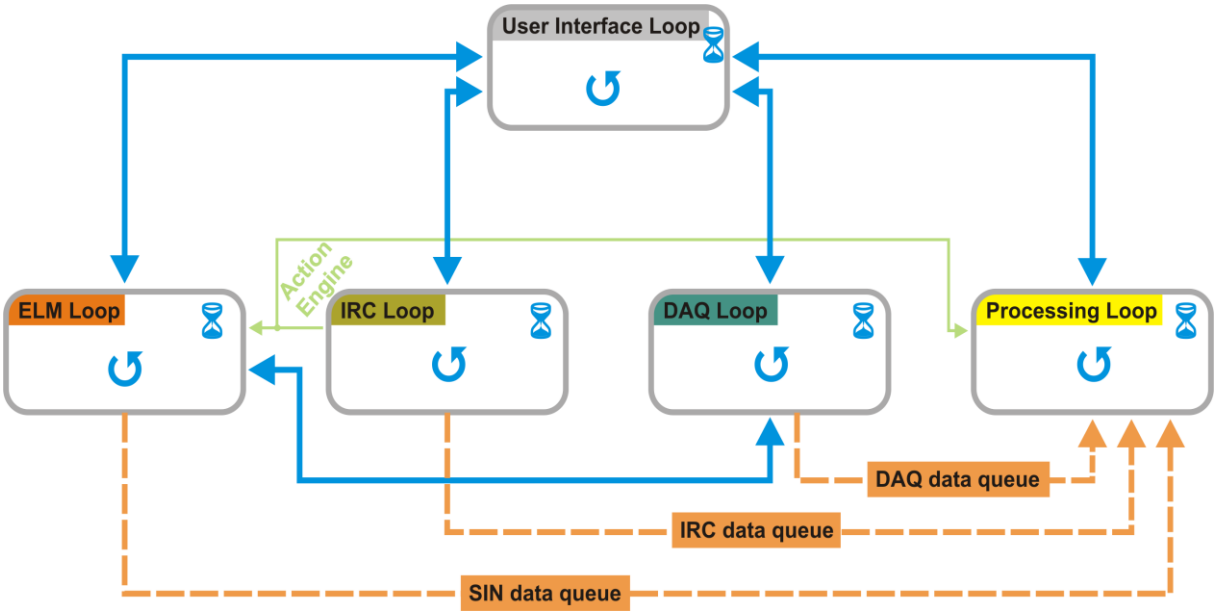


Figure 7.1 – DREAM application top-level flow diagram.

The created concept as it was described has other advantages. Its modularity is important. It is expected that in future some parts of the apparatus can be replaced. The program is designed so that any part of it can be altered, replaced, or upgraded without affecting the other parts and with minimum effort. This can be achieved by adding or changing states in particular QSM or by replacing/adding the whole QSM. The ELM Loop was programmed using the object-oriented approach to allow future quick implementation of any communication protocol to drive the connected electromotor.

8.1 UI Loop

The UI (User Interface) loop handles the user interaction and establishes communication with the other loops. Its main performed tasks are:

- The application parameters are read from the configuration file during the application boot up
- Initializes controls
- Handles user interactions
- Scales the graphs, controls
- Establishes communication with the other QSMs
- Feeds the other QSMs with parameters they need to operate
- Assures central error handling

In case of errors, it sends commands to the other queues to stop them and to safely stop the electromotor and engine. If any error occurs in any of the loops they immediately inform the User Interface loop.

8.2 ELM Loop

The ELM Loop is fundamental, since it handles communication with the controller that drives the electromotor. More importantly, if selected, it compensates for the influence of the slip of the ribbed belt (used because of safety reasons, see chapter *3.1.2 Electromotor, inverter, control unit, gearbox*) which is used for the torque transition between the shaft of the electromotor and the shaft of the combustion engine. To compensate for the slip, it obtains the data from the IRC loop in order to get the actual value of the shaft rpm and to be able to compute necessary corrections to reach the exact desired rpm. The speed adjustment is performed after the electromotor finishes ramping and revolves at constant speed.

The data between the IRC QSM and ELM QSM are shared through the Action Engine. In this case the AE works as a circular buffer storing the means of each of the last *n* revolutions (*n* is a preselected amount of revolutions). The mean is computed since the speed of the shaft during one revolution fluctuates (due to engine's rotational irregularities). When queried, the AE computes the median of the stored means to get the actual shaft rpm. This way possible false readings and extreme values are eliminated.

Multiple-level safety is assured by watching the crucial parameters:

- The amount of slip correction steps is limited (by default to 10 steps – adjustable via the configuration file).
- The maximal difference allowed between the crankshaft-converted IRC speed and the crankshaft-converted electromotor speed is limited (20%). The difference is given by the slip of the ribbed belt. The term *crankshaft-converted* means that the speed of the particular shaft is converted to the speed of the crankshaft. The conversion is done with respect to the actual gearing between the shaft and the crankshaft.
- The application recognizes if the maximal frequency allowed by the inverter (p1082) was reached.
- It employs a watchdog timer (if supported by the controller) to stop the electromotor if the controller doesn't receive a telegram within a specified time (2 sec). It handles the situation like a communication cable disconnection or unexpected application hang-up.

The ELM Loop was coded using the LabVIEW object-oriented programming (LVOOP). This approach enabled us to implement and choose on the fly between multiple communication protocols to drive different electromotor types. We implemented the Siemens USS protocol (for the TUL apparatus) and the LENZE LECOM protocol (for the ŠKODA AUTO apparatus). If a different electromotor with different communication protocol is to be used with the developed application, changes in the application itself will be minimal.

8.3 IRC Loop

IRC QSM continuously performs measurement of speed and speed fluctuations of the revolving shaft (typically camshaft) utilizing the IRC sensor and the counters of the counter board. The selected methods and the measurement uncertainty are discussed in chapter *5 Frequency measurements.*

With higher rpm (even with the advanced application architecture) it would be risky to read data from one revolution at a time since it might not be fast enough. For this reason we always flush the whole DMA buffer and separate only the data representing the whole

periods, keeping the redundant part for next iteration. Read speed array is sent for further processing: moving average filtering and order analyses.

Besides the frequency measurement the loop performs additional counter operations as described in the chapter *6 Additional counter tasks.*

8.4 DAQ Loop

This loop performs the data acquisition of the valve displacement and velocity depending on the cam displacement. The IRC TTL pulses are used as a sampling signal. The temperature, pressure of the oil or other signals can be also measured. If needed, the sensors can be supplied with power directly from the multifunction card.

The data are read continuously and if the desired rpm is reached (the ELM Loop informs) the next dataset is flagged and tested in the Processing Loop for signal drop-outs, offset drift and other phenomena. If the tests are successfully passed, the transition to the next rpm is started (the ELM Loop is informed).

8.5 Processing Loop

The Processing Loop processes the data from the data queues. Since the human eye has limitations, we don't necessarily have to refresh the indicators during each revolution of the shaft (which would be impossible anyway with higher rpm). This loop always flushes the data queues, processes them, saves the data flagged for saving and displays only the last available data sets. It runs on its own speed without influencing the rates of the other loops. Its performed tasks are:

- Processing of data from the data queues
- Graphs updating
- Testing of the measured data of the valve displacement and velocity
- Data saving

The signal drop-out recognition tests are the core of the Processing Loop. If the measured data at a given speed are affected by signal drop-out, it will perform operations to assure the measurement repetition. This way only the high-quality data are saved. The algorithm and the motivation of its development is described in the chapter *9 Algorithm for signal drop-out recognition in IC engine valve kinematics signal measured by laser Doppler vibrometer.*

List of the tests programmed:

- Signal present tests
- Signal offset drift tests
- Signal false trigger recognition
- Signal drop-out tests

The tests are performed on the valve kinematics signals in multiple data intervals with different parameters for each interval. A total of 13 tests are done on each read period. The tests themselves were programmed using the LVOOP, thus adding of new tests in the future is very simple.

9 Algorithm for signal drop-out recognition in IC engine valve kinematics signal measured by laser Doppler vibrometer

We present an algorithm for the recognition of signal drop-out developed particularly for measurements of valvetrain kinematics. This algorithm is needed in order to save data that are not affected by a drop-out phenomenon and thus allow full automation of the measurement. Such an algorithm will increase the throughput of the engine test stand and decrease the time needed for the evaluation of the valvetrains of combustion engines. The work shows the most commonly encountered drop-outs and their characteristics and locations. It presents an automatic separation algorithm for the measured signal so that the drop-out recognition tests can be aimed at specific data intervals (valve opening, valve closing, etc.) with specifically set parameters of the algorithm.

9.1 Motivation for the drop-out recognition

High speed Laser Doppler Vibrometry (LDV) has become a standard measurement technique for obtaining the kinematics of the valves [6], [8]. The main reason is that the technique is non-contact and offers information about both the valve displacement (using the fringe counting technique [9], [10]) and the valve velocity (based on the Doppler effect) up to high engine speeds. The LDV technique also has its drawbacks. It is sensitive to dust and oil droplets that might appear during the measurement and demands precise focusing of the lenses. However, the main problems arising during the automation of the measurement are *speckle noise* and *signal drop-outs.*

A speckle pattern is produced when the coherent waves of the incident laser beam are dephased during backscatter from a surface that is rough on the scale of optical wavelength. The scattered yet still coherent waves interfere constructively and destructively, producing a chaotic distribution of light and dark spots [11], [12]. For the measurements of the valve kinematics the valve is equipped with a retro-reflective tape to achieve higher intensity of the backscatter light. Unfortunately, the tape itself is optically rough and still produces a speckle pattern [12], [13]. The speckle pattern is not of high significance unless it changes dynamically. Then it can translate (i.e. speckles appear to move in space while retaining their size and shape) or boil (i.e. no translation of the speckle but a continuous evolution from one size and shape to another) [41], [42]. The speckle noise is produced if the Doppler signal amplitude remains high enough for the demodulator to operate. If the Doppler signal amplitude drops to low levels and the demodulation process fails, signal drop-out occurs. During its motion the valve is designed to rotate in order to keep the valve face and seat clean of carbon deposits. This also has the effect of slightly reducing the wear [43]. Because of the acting forces the valve also experiences tilt especially at high speeds. This altogether contributes to significant speckle noise during the valve kinematics measurement and frequent drop-out noise.

Attempts have been made to define strategies for reducing the signal drop-out noise to a manageable level for industrial diagnostics [44]. Algorithms for automated identification of the presence of the signal drop-outs and the selection of an unaffected portion of data were presented for measurements on bearings of electric motors [45]. However, the valve kinematics data are very specific so we decided to develop our own algorithm for signal drop-

out recognition. To achieve this, it was first necessary to study data measured over the last 20 years to clearly identify where the drop-outs occur and to characterize them.

Since all presented figures depict data of real engines where some are still on the market, we decided not to show the real data scales but rather normalized values. As shown later, the validity of the described algorithm isn't compromised, especially because the normalization is its first step.

9.2 Phenomena affecting the valve kinematics signal

Besides the signal drop-outs, other phenomena affecting the measured signal were identified. In this chapter all of the encountered problems are described. However, in the following chapters focus is given to the drop-out phenomenon since it is the most common.

9.2.1 Signal drop-out during valve opening or closing

In measured data (valve displacement or velocity), one-point peaks that apparently do not form real measured values can be encountered (Fig. 9.1). They are the afore-mentioned signal drop-outs resulting from failed signal demodulation. The vibrometer controller is equipped with so-called tracking filter [24] whose purpose is to bridge those instances and improve the signal-to-noise ratio of the weak optical signal. Nevertheless, the filter has its limits. The signal drop-outs were identified while using different laser units (Polytec OFV 502 + OFV 3000, Polytec HSV 800 FF + HSV 2000) and with different measurement equipment and settings as well as in different laboratories. The better focus of the probes was performed and kept during the measurement, the less drop-outs in the measured data were encountered. The cleanness of the retro-reflective tape was also important. The size of the drop-outs ranges from short and hardly-recognizable peaks to very high and clearly visible.

Other source of undesired peaks can be an oil drop crossing the path of the laser beam. In case of oil or any other particle we can observe its effect on both of the measured curves (Fig. 9.2).

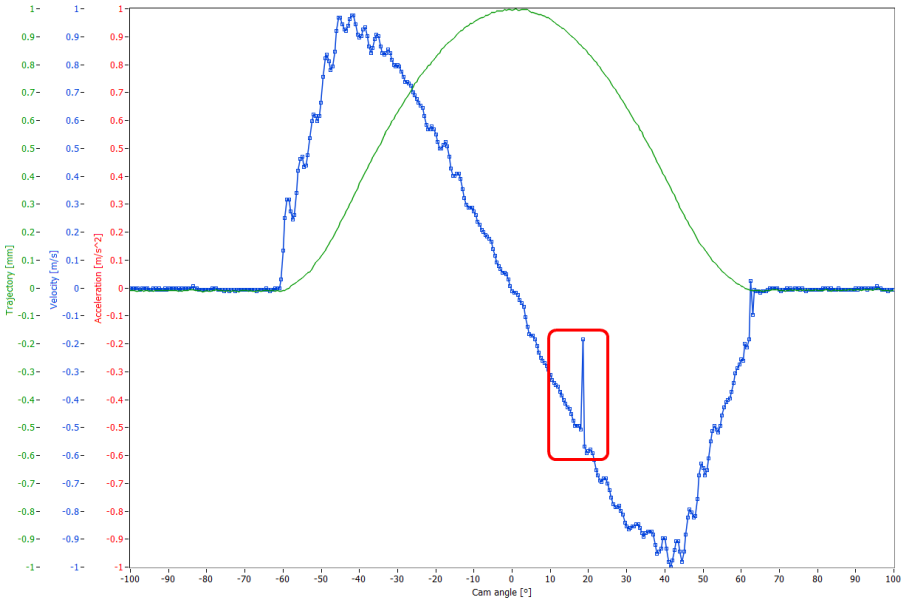


Figure 9.1 – Velocity signal drop-out

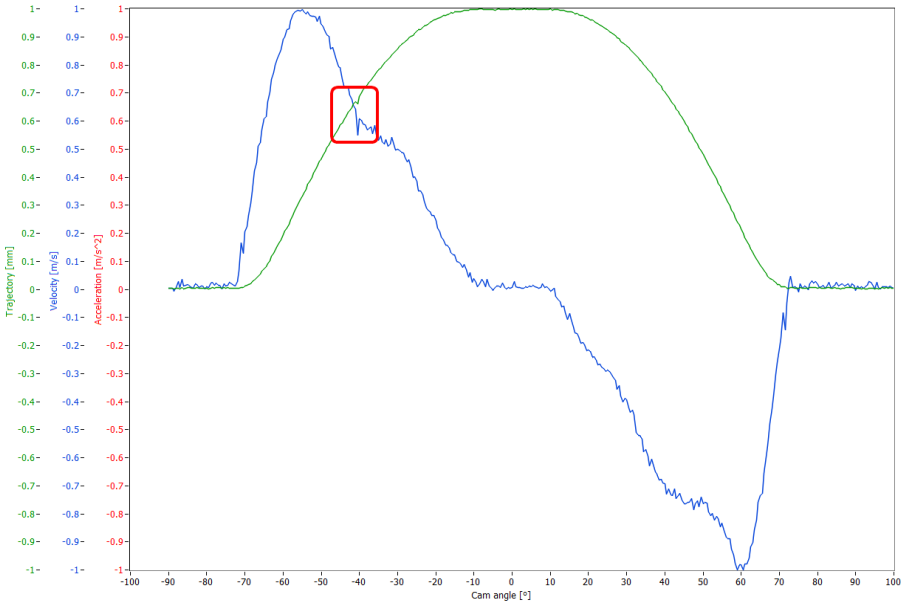


Figure 9.2 – Laser beam crossing

9.2.2 Base circle signal drop-out

Same principle as in *9.2.1* but it occurs on cam's base circle (Fig. 9.3). It can happen in both signals. To decide whether the peak is or is not a signal drop-out largely depends on the noise level of the measured curves. The same peak if surrounded by noise would not be considered a problem. The size is significantly lower than in *9.2.1*.

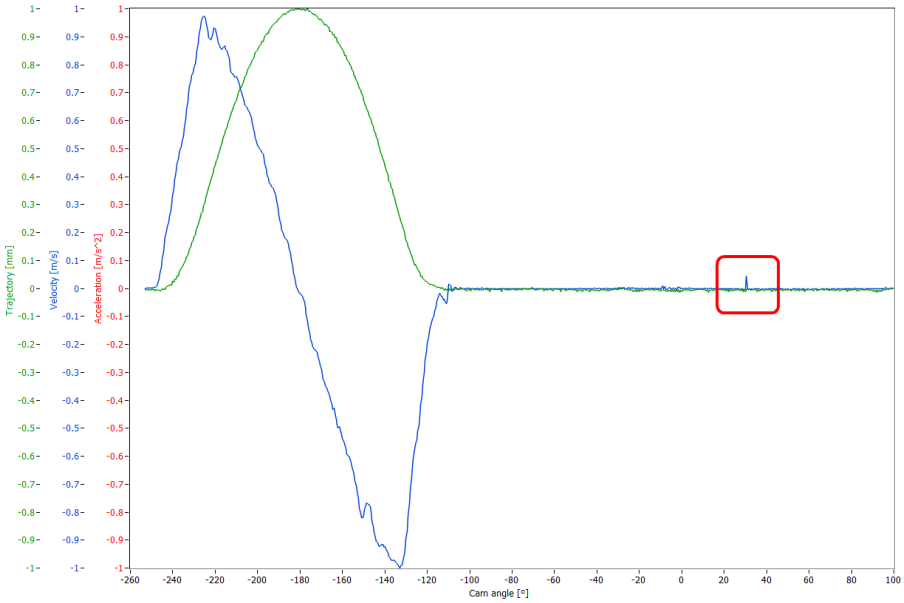


Figure 9.3 – Base circle signal drop-out

9.2.3 Offset drift

Inaccurate settings of the devices can result in offset drift (Fig. 9.4). When the valve is located on the base circle, the expected value of both velocity and displacement should be 0 V. Furthermore, the offset drift can be encountered due to a false reset pulse arrival. The pulse is used to reset the interferometer to assure that on the base circle the value of the measured signals will be referenced to 0 V. The IRC trigger signal is used for this purpose. The false arrival can occur due to vibrations or magnetic interference from the electromotor.

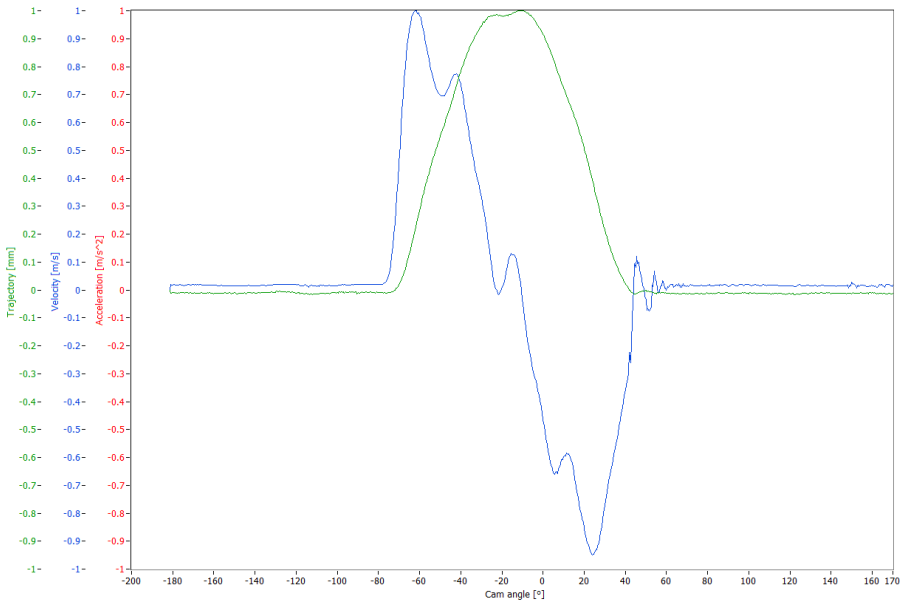


Figure 9.4 – Velocity offset drift

9.2.4 Complete signal distortion

Resulting signal does not resemble the expected measured data (Fig. 9.5). This mostly happens if focus of the probes is lost (due to vibrations or another major problem).

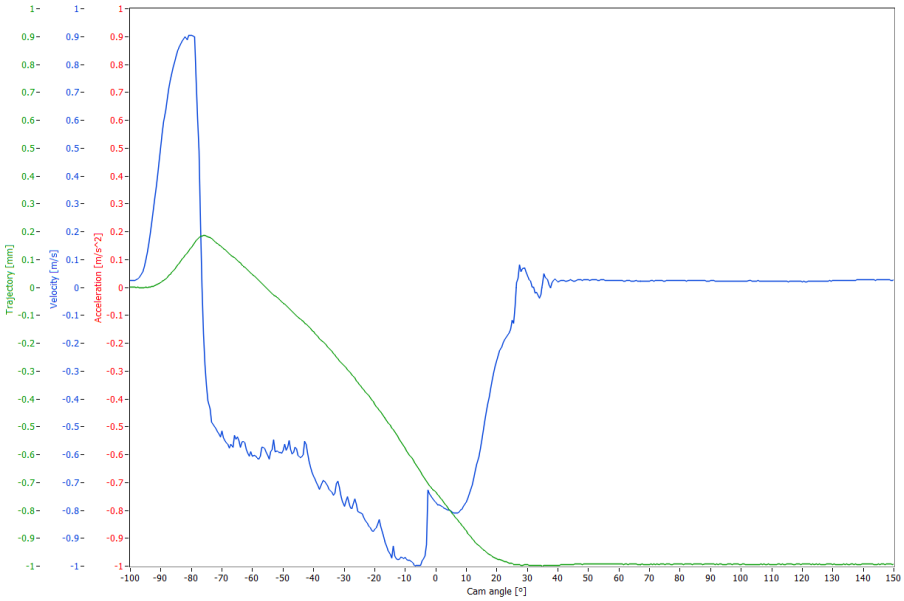


Figure 9.5 – Complete signal distortion

9.3 Localization of the phenomena

Localization of each of the named problems in the measured data is important. The parameters of the algorithm can be better targeted if we narrow the intervals for testing. This will improve accuracy and time needed for testing.

The measured data were divided (Fig. 9.6) into five intervals (*T1-T5*) based on the occurrence of the phenomena listed in the previous chapter. Significant endpoints were marked in the data.

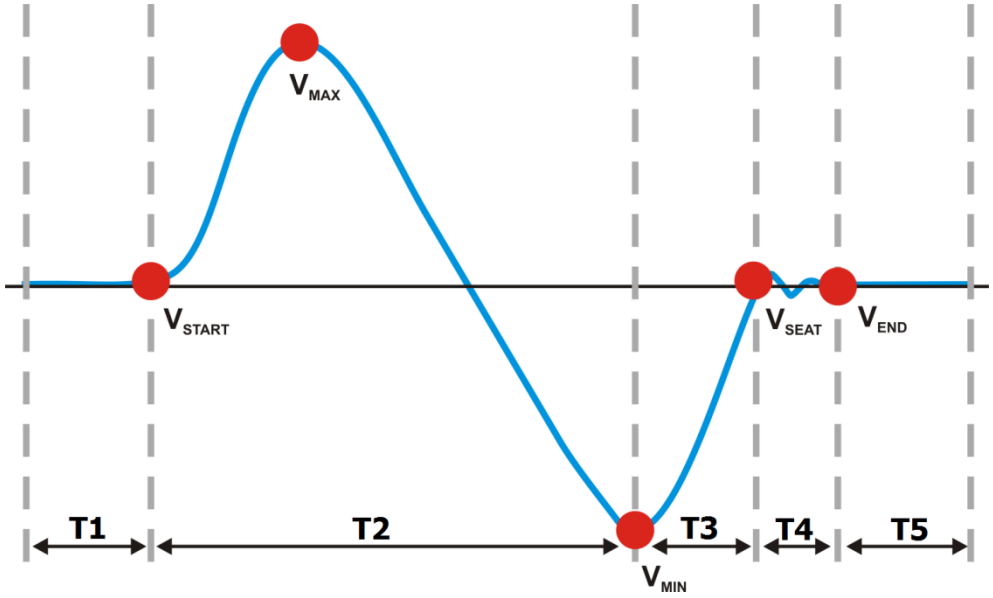


Figure 9.6 – The data intervals

Interval T1 is the portion of the signal before the valve opens. The valve is located on the base circle of the cam. In this interval offset drift of the displacement or velocity can be

detected. Also the signal drop-out of both of the measured variables can occur. The left endpoint is given by the trigger pulse that starts the data acquisition. The right endpoint is the point V_{START} where the valve begins to open.

Interval T2 is the portion of the data where the occurrence of the velocity drop-out can be observed. This is the most common case and location of its occurrence. The left endpoint of this interval is the point V_{START}. The right endpoint is the minimum of the velocity (i.e. maximum during valve closing) marked V_{MIN}.

Interval T3 is very sensitive. Only its first half should be tested for the velocity signal drop-outs and the test parameters should be set to identify only drop-outs of significant size. In the second half the first contact of the valve with the seat occurs which poses a step change of the velocity and could be easily mistaken for the signal drop-out (for comparison see Fig. 9.7). A typical example would be a valvetrain with worn valve guides. In such a situation the algorithm would re-start the measurement, thus eliminating the possibility to discover the malfunction of the valvetrain. The second half of *T3* could be used in the future for detailed study of automatic determination of the impact velocity, which is a crucial parameter for the valvetrain lifetime.

Intervals T2+T3 together form interval where the test for detection of displacement signal drop-outs should be applied. In this case the test can be applied also to *T3* since the valve's first contact is not detectable from the displacement curve and thus cannot be mistaken with the signal drop-out. The right endpoint V_{SEAT} is a theoretical point where the valve fully sits for the first time in the seat. As will be shown later, this point is easily determinable.

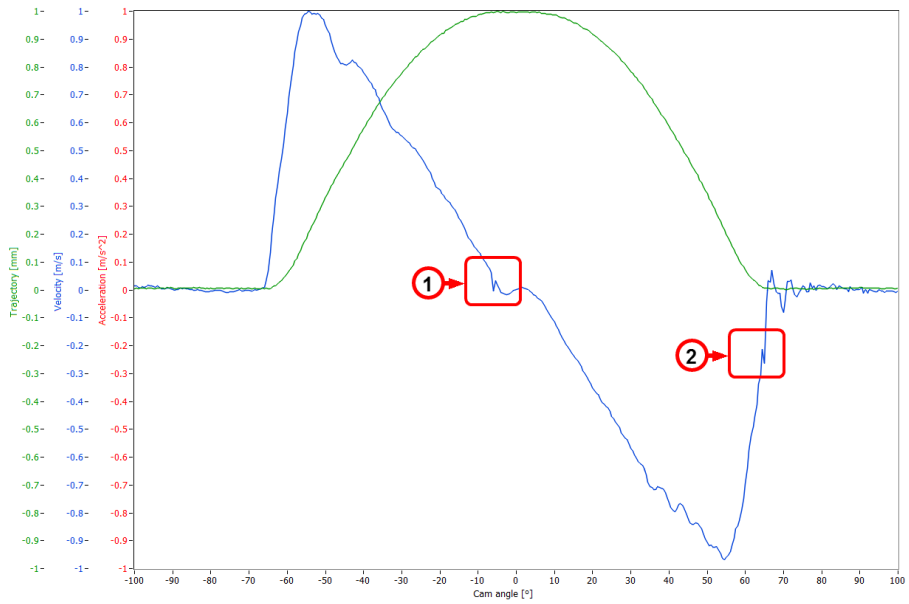


Figure 9.7 – *1* denotes signal drop-out. *2* denotes first contact of the valve with seat

Intervals T4+T5 together should be used for detection of the displacement offset drift and displacement drop-out. The left endpoint is V_{SEAT} and the right is the end of the data record.

Interval T5 is used for the velocity offset drift detection and the velocity signal drop-out detection. *Interval T4* is not tested, since there the valve can still oscillate with step changes in velocity. The left endpoint is given by point V_{END} which marks the point where the valve does not oscillate anymore (the valve is on the cam base circle and does not bounce).

These intervals denote portions of data that will be introduced to particular tests. As mentioned earlier, the velocity and displacement of the valve are measured. For separation into the described intervals, the velocity curve is used. It is not possible to separate the

measured data into meaningful intervals based on the displacement curve, since the changes in displacement are too small and do not allow for clear identification of each phase of the valve cycle. It is not possible to use the acceleration either. It is obtained as finite differences from the velocity and has very noisy character. Multipoint difference methods could be used or smoothing of the signals could be applied. However, since different noise levels are present across the measurements and different pprs are often utilized, it would be difficult to create a smoothing algorithm suitable for every condition.

9.4 Separation

It is crucial for further processing to precisely identify the intervals' endpoints. It is not a trivial task since their determination can already be influenced by the existence of phenomena like offset drift or signal drop-out. Different resolutions of the data acquisition that can be used also play a role. All of the following parameters of the separation algorithm and parameters of the resulting drop-out detection tests are listed in the *Appendix 4 – Parameters of the drop-out recognition algorithm.*

The first step to successfully determine those points is to normalize the acquired data. A normalization that provides a unified range of values for our algorithm has to be chosen. This is because the shape of the measured curves can vary significantly across different valvetrains, as can the absolute values across different speeds. The key seems to be the *normalization to "1"*. V_{MAX} and $abs(V_{MIN})$ are identified in the velocity data. The higher of those two is chosen and the whole dataset is divided by this value. This way the velocity in the range of -1 to 1 m/s is obtained.

During the measurement, five continuous datasets for each engine speed are acquired. Each dataset is independently analyzed and V_{START}, V_{MAX}, V_{MIN}, $V_{SEAT,}$ V_{END} for each of them is determined. Then the median value is taken, thus filtering out extreme values and obtaining coordinates of each endpoint for a particular engine speed.

V_{MIN}, V_{MAX} – these two points are easily detectable. They are the maximum, respectively minimum (maximal velocity during valve closing) of the velocity.

V_{START} – is the point where the valve begins to open. It is important to realize that a simple condition such as: "*We look for a point where successive N points lie above zero value*" won't be enough. This approach fails if the velocity has drifted offset and it will find false index. For our purposes following definition seems to work quite well: "*We look for an index where all points between this index and the velocity maximum have a value higher than the value of the point on that index."*

The second necessary condition utilizes a simple outlier test [46]. Such a point and the following 3 points (generally *p1* points, where *p1* is a parameter of the algorithm) has to have the deviation from the mean of all the preceding points at least 3 times (generally p_2 times, where p_2 is a parameter of the algorithm) higher than the standard deviation ('three-sigma' rule [47]) of the preceding values. In other words, point where the curve starts to rise steeply is found, e.g. the point where the cam profile changes from the base circle.

In summary:

We take the values of velocity $v_1 \dots v_{MAX}$ and we look for v_i so

i. $v_i < v_j \; where \; j = i+1, \; i+2, \dots MAX$

ii. $\sigma_{i-1} < p_2 \cdot s_k \; where \; k = i+1, \; i+2, \dots p_1; \; s_k = \sqrt{(v_k - \bar{v}_{i-1})^2}$

$\bar{v}_{i-1}$ is the mean of the preceding values $v_1 \dots v_{i-1}$ and σ_{i-1} their standard deviation.

V_{SEAT} – is the point where the valve reaches the seat the first time. Especially with high engine speed the valve can bounce a few times [6] or oscillate before it fully closes and remains on the base circle of the cam. Such a point is easily detectable by examining the data of the velocity from the V_{MIN} index to the end of the dataset. We attempt to find a point that has a value higher than the mean of the rest of the consecutive data, e.g:

We take the values of velocity $v_{MIN} \dots v_m$ *and we look for* v_l *so* $v_l > \bar{v}_{l+1}$ *where* $\bar{v}_{l+1}$ *is the mean of the values* $v_{l+1} \dots v_m$

V_{END} – is the point where the bouncing of the valve (if present) disappears and the valve fully follows the cam base circle profile. The ideal situation would be $V_{END} = V_{SEAT}$ which would mean the valve smoothly closed without bouncing. The data from the index of V_{SEAT} to the end of the record are taken. The mean of this portion is calculated and the oscillations are then analyzed. The highest value (amplitude) of each part above and below the mean level (Fig. 9.8) is determined.

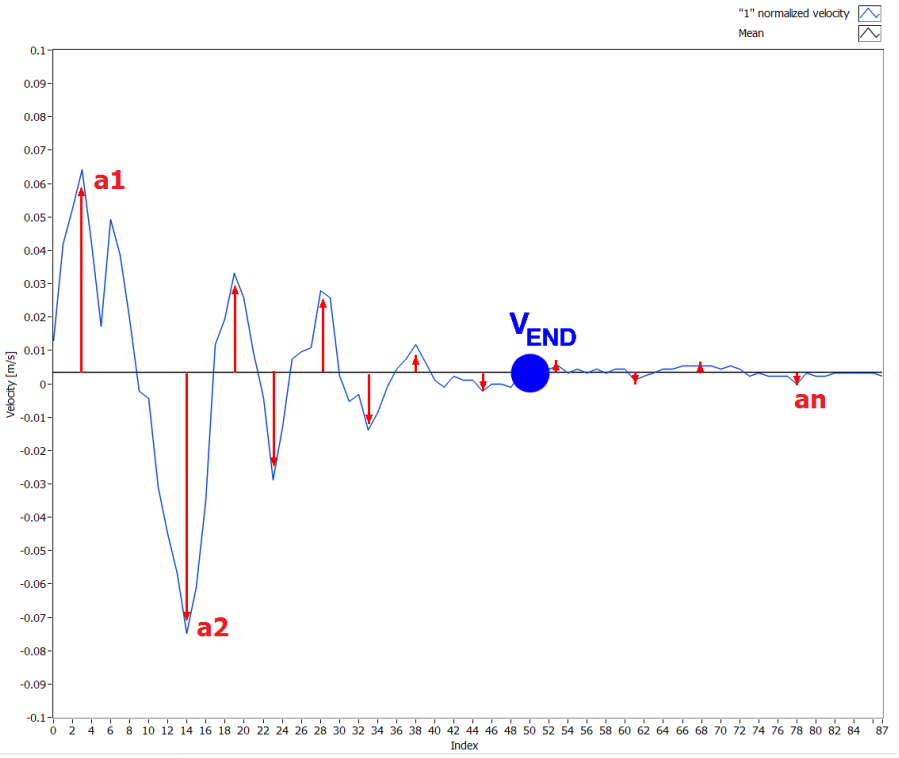


Figure 9.8 – Determination of V_{END}; a_n = amplitude of oscillation

The average amplitude of the oscillations $\bar{a}$ is calculated. The algorithm then looks for p_3 consecutive sequences that have the amplitude less than the given threshold. Parameter p_3 is dependent on the measurement ppr. For example for ppr = 720 it looks for 4 consecutive sequences meeting the criteria. Every additional 1800 ppr raises the number by 2. The threshold is set as p_4 times $\bar{a}$. If the calculated threshold is higher than parameter p_5, p_5 is used instead. This can happen if the dataset ends before the oscillations disappear (usually if the DAQ was started too early on the base circle right after the valve closed).

9.5 Tests

Tests to automatically recognize the signal drop-outs in the above described intervals were developed. The tests have to be accurate and robust at the same time. This will assure that the tests won't fail even with very special valvetrain designs, such as racing valvetrains. At the same time the tests have to be able to distinguish between the phenomenon and the real trait of the valvetrain. Let's assume a situation that the examined valvetrain has a malfunction that appears only once in a while. If the tests weren't designed accordingly they could identify the malfunction as the signal drop-out. They would repeat the measurement until they'd get a satisfactory result. This way the engineers would never realize that the valvetrain deserves a detailed investigation.

The test of the velocity signal drop-out recognition for *Interval T2* will now be described, along with the process used to achieve its final state. The other signal drop-out tests are merely a variation with slightly different parameters.

9.5.1 Signal drop-out test

The first step of the algorithm for the drop-out detection is *"1" normalization*. The second is to obtain *Interval T2* by determining its endpoints.

The following approach we took at the beginning and later abandoned was to examine the acceleration. As can be seen (Fig. 9.9) acceleration is not always helpful. The marked peak is clearly visible in the velocity but is hard to detect in the acceleration. The left side of the picture shows acceleration calculated from velocity by forward difference method $\Delta f(x)$. The right side is the same signal but the data were smoothed by Savitzky-Golay filter [48] with parameters: local polynomial regression of 3rd order, 5 points series.

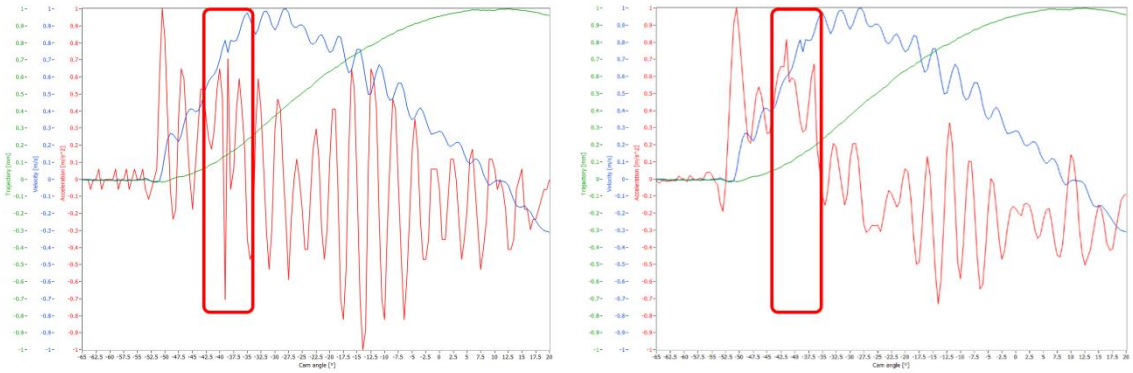


Figure 9.9 – Drop-out in the velocity and corresponding acceleration change. Left: $\Delta f(x)$; Right: $\Delta f(x)$ + Savitzky-Golay filter (3rd, 5 points)

The other considered procedure was to conduct a comparison between the datasets acquired at the same engine speed. By comparing them against each other it would be possible to identify whether a signal drop-out appeared. Unfortunately even slight phase drifts do not allow the algorithm to become very accurate. Also the danger of identifying valvetrain malfunction as signal drop-out was too high with this approach.

The next thing that was considered was to utilize Short-time Fourier transform (STFT). In theory, such a peak of signal drop-out would show unusually high frequency. However, for this purpose the measurement resolution was not sufficient.

We also tried to use some of the conventional cubic spline techniques used in ECG processing [49] to remove baseline wandering. With this approach it would be possible to remove the baseline and focus only on the present drop-outs. The cubic spline method does not suit the valve kinematics data very well and the algorithms always had to be adjusted for a particular dataset based on its noise level, ppr of the DAQ and type of valvetrain. Thus it was difficult to estimate the level of fitting.

Three other techniques were then compared (Fig. 9.10, 9.11). Since the signal drop-out peak always consists of only one point, the first technique used was to calculate the distance from the center of the two points on the sides. This way, a graph of extremes on the baseline was obtained. The second approach was to utilize the Discrete Wavelet Transform (DWT). The first level of the *detail coefficients* was used to recover the high frequency portion of the data. Different types of wavelets were tested. The *db03* wavelet (which is asymmetric, sharp, irregular and resembles the shape of the peak) seems to show the best results. The third approach was to use a highpass digital FIR filter designed for our purposes. The Parks-McClellan [50] algorithm (SB=200Hz, PB=4000Hz) was used to obtain the filter coefficients. Since each filter has some start-up time, it was necessary to avoid the artifacts at the beginning and at the end. The data were symmetrically appended by reversing the needed amount of samples (which equals the order of the filter) at the beginning and the end. The data were then filtered for the first time. The filter parameters were set as if the data were sampled with constant frequency F_s=10 kHz. This way, universal settings for filtering out the baseline could be set (the cutoff frequency SB/LB won't change with different ppr and the amount of samples in the interval under examination). Then the data were reversed and filtered again with the same settings. This way the shift introduced by the filter was removed. In the end the data were reversed back and the data portion that was appended to the beginning and the end was deleted. The absolute value of the resulting signal was taken in order to clearly show the peaks.

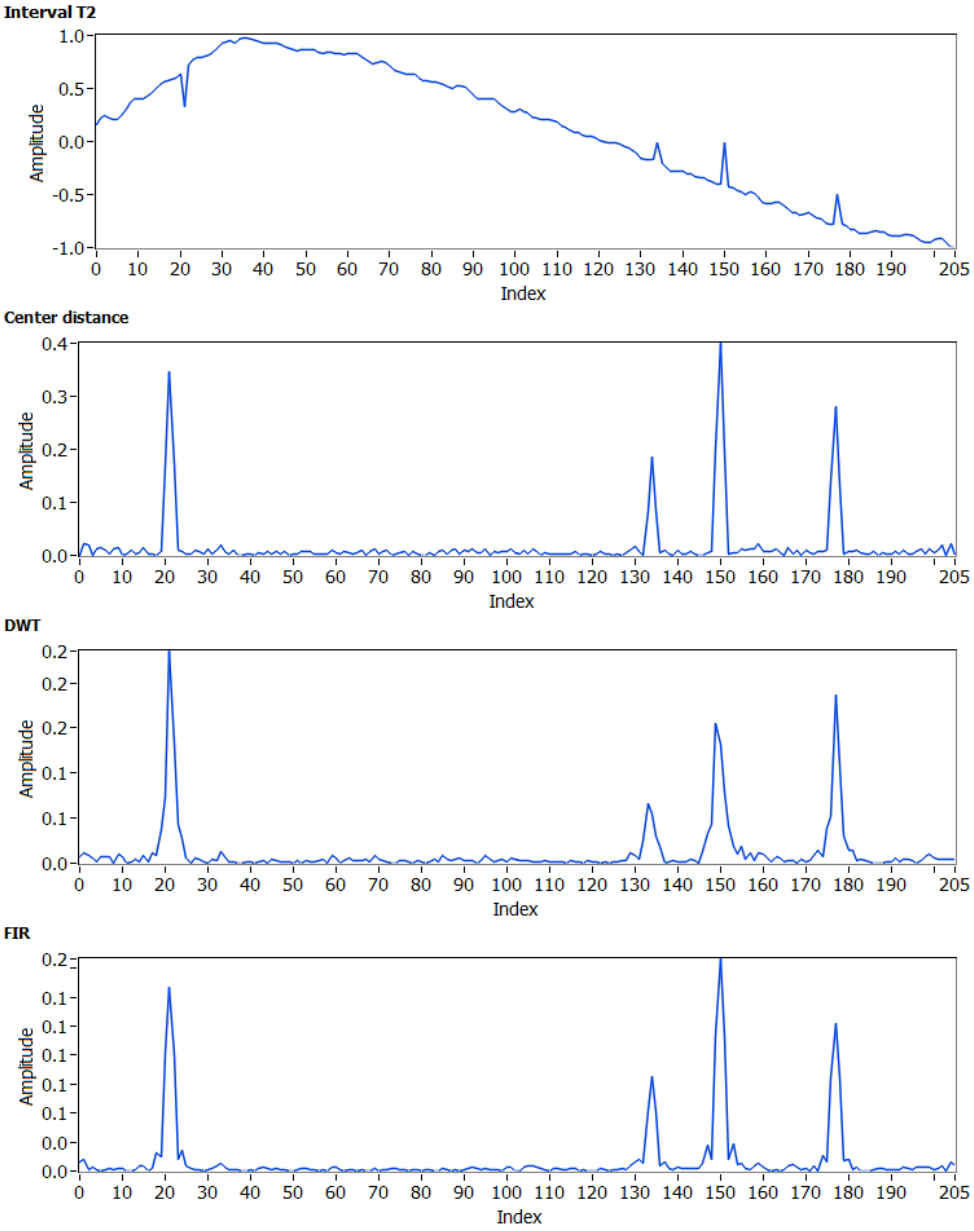


Figure 9.10 – Different techniques of signal drop-out detection

From those three techniques the custom FIR filter approach was chosen. It delivers the best results and the best signal-to-noise ratio (SNR) even for small peaks as can be seen in (Fig. 9.11)

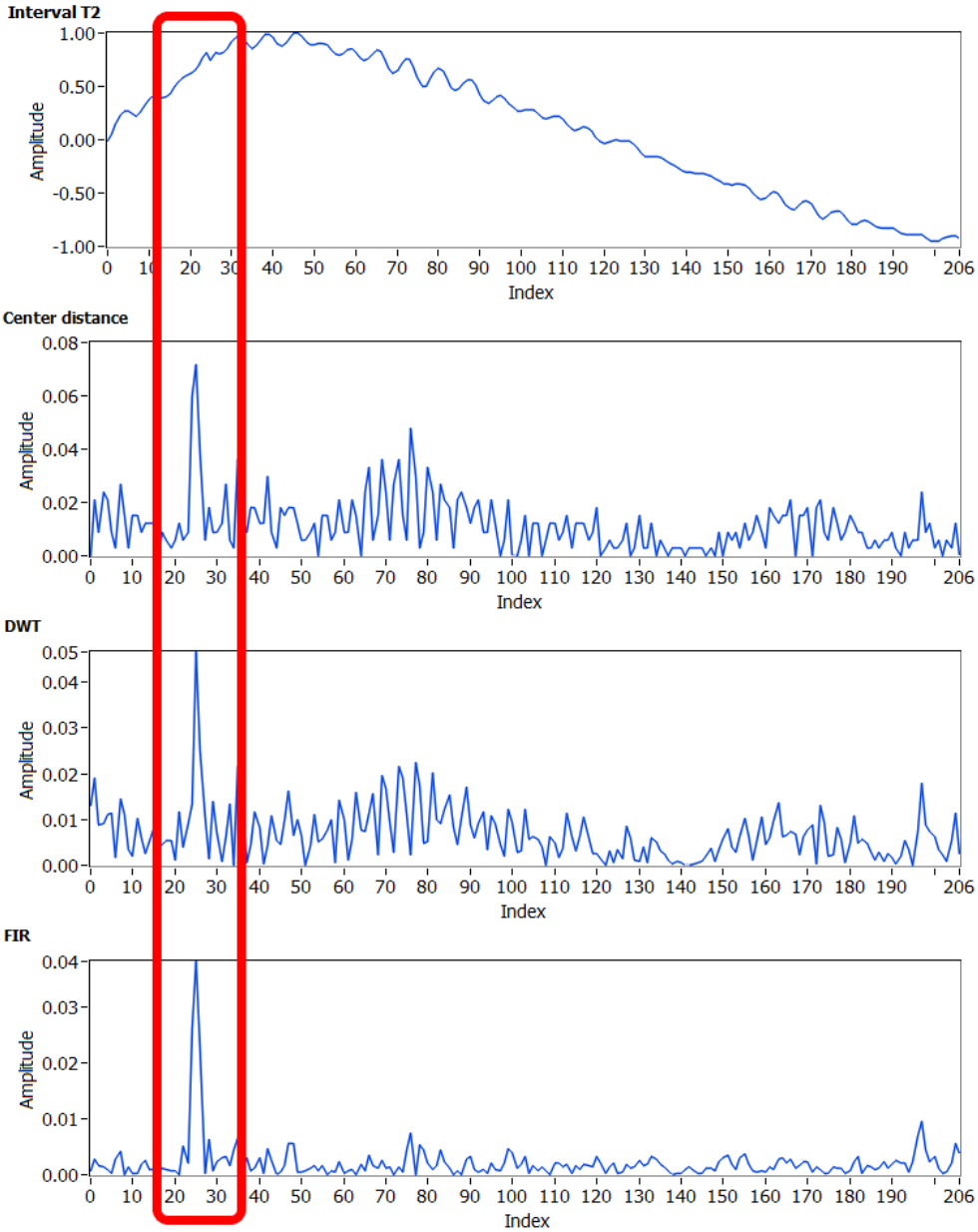


Figure 9.11 – Different techniques of signal drop-out detection; small peak

Now the signal (which may or may not contain drop-outs) is available. The next step is to recognize whether it was affected by the phenomenon. Simple thresholding would not be sufficient. The same peak which can be considered drop-out in one dataset would not meet the criteria in another. What distinguishes them is the level of surrounding noise. This situation is clearly visible in Fig. 9.12. In the top picture the peak reaches the value 0.04 and should be marked as signal drop-out. In the bottom picture the same value would not be considered a signal drop-out. The red line shows the estimated noise level.

If the noise level of the signal can be determined, the SNR of the data can be calculated by dividing the value of the top of the highest peak (h) by the noise level (n). The SNR has become the criterion which decides whether the measurement should be restarted.

$$SNR = \frac{h}{n} < c \qquad \text{where } c \text{ is the criterion.} \tag{9.1}$$

If SNR > c than the signal should be re-measured.

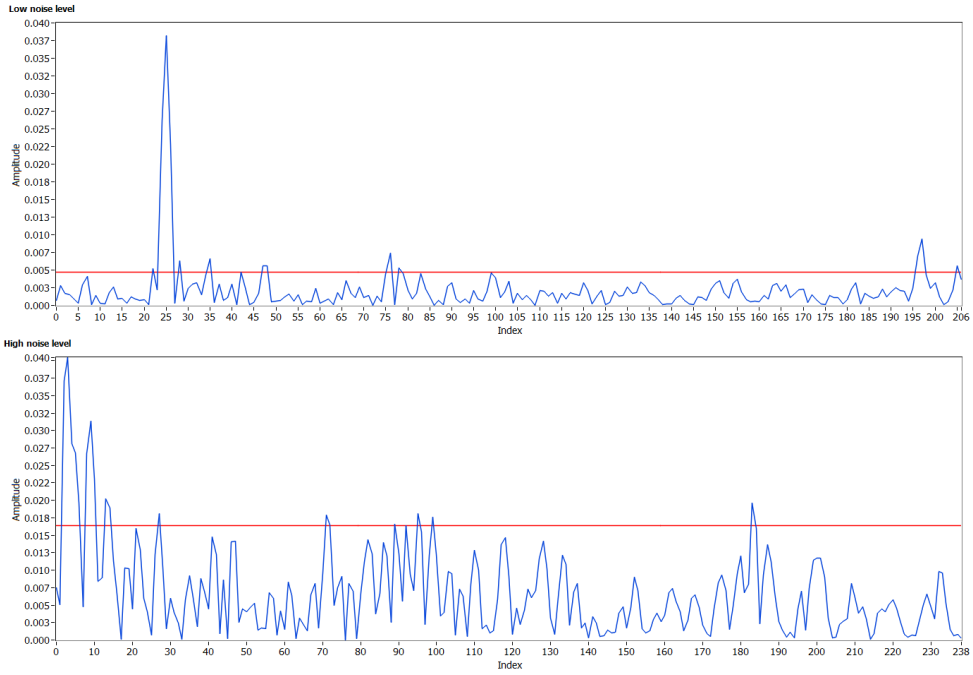


Figure 9.12 – Comparison of low noise level and high noise level

To determine the noise level a statistical approach was adopted. Our data after filtration consist mostly of peaks of speckle noise that was superimposed on the baseline. The occurrence of the signal drop-outs in one dataset is not high (usually one peak in one dataset). The occurrence of multiple drop-outs is quite rare and the situation depicted in Fig. 9.10 is the worst possible scenario.

First the peaks are sorted in ascending order by their size. Then the peak that is at 85% of the total count of the peaks is selected (in other words the 85th percentile). Its value determines the noise level. This approach is possible since the drop-out peaks will always be among the highest 15%. Fig. 9.13 shows the described procedure with the peak shown in Fig. 9.11.

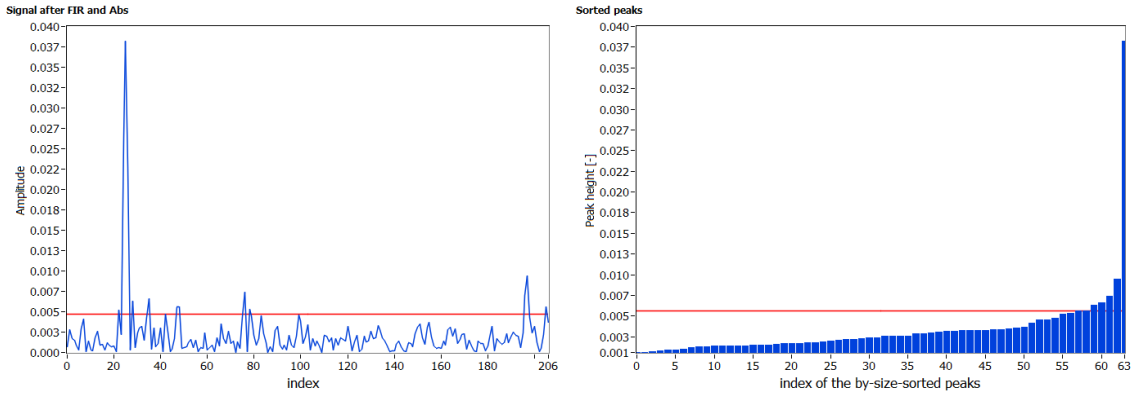


Figure 9.13 – Peak sorting and 85% noise level estimation

The parameters of this algorithm are then the percentage used to determine the noise level (*p6*), the SNR criterion *c (p7)* and also the minimal size of the drop-out peak h_{min} *(p8)*.

For the velocity signal drop-out detection algorithm for *Interval T2* the parameters after testing were set as $p_6 = 85\%$, $p_7 = 4.5$, $p_8 = 0.01$. In other words, to mark the dataset under investigation as affected by the drop-out phenomenon, it has to have the SNR of 4.5 or higher and the highest peak has to be higher than 0.01. The noise level is obtained by taking the value of the 85% peak.

To keep the robustness of the detection algorithm, an additional evaluation of the provided results was implemented. If the algorithm marks more than half of the datasets taken at the same engine speed as faulty, the evaluation algorithm will check the positions of the detected peaks. If the positions of the drop-outs are the same (or nearly the same) then the signal is saved so the engineer can decide whether the data are showing a trait of the valvetrain or just a drop-out. The p_9 parameter of the algorithm is the allowed peak index position difference. It is expressed as the percentage of ppr.

The full flow diagram of the algorithm is depicted in Fig. 9.14.

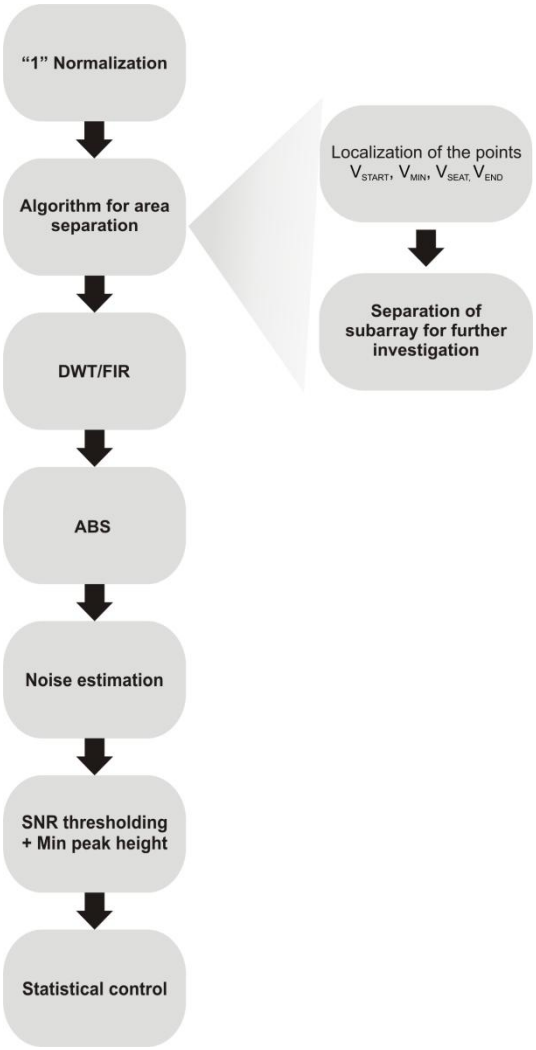


Figure 9.14 – Data flow diagram of the algorithm for signal drop-out detection

9.6 Summary

We created a set of routines which were aimed at the recognition of signal drop-outs in the valve kinematics data. The developed algorithm should help the engineers to increase the

throughput of the engine test stand and decrease time needed for the evaluation of the valvetrains.

The routines were tested on data which were collected for over 20 years and on data measured on combustion engines in our laboratory. The tests show satisfactory results in the identification of the phenomenon. Small drop-outs which are recognizable by the human eye might be on the edge of the algorithm's capabilities, especially if their size is not far from the estimated level of noise. In such a situation the tests would most likely fail. Another obstacle is to unambiguously decide what is and what is not a signal drop-out.

Future work should be aimed at monitoring this algorithm in everyday testing for an extended period of time, and adjusting its parameters. There is also high demand for the creation of an advanced algorithm for the automatic identification of impact velocity, which poses a great challenge.

10 Comparison of measured data of two different valvetrains

To verify the capabilities of the constructed system, a series of measurements was carried out. First the standard 1.2 HTP OHC valvetrain in the half-engine setup was measured. Then it was replaced with a low friction version of the valvetrain and measured again. The low-friction type has thinner springs with smaller stiffness; the cam lobes are thinner as is the cam follower to achieve smaller contact area and less mass. Different materials and bearings are used to achieve smaller rolling friction. The valves are the same in both cases.

All the measurements, unless stated otherwise, were carried out from 1000 rpm to 5000 rpm with an increment of 1000 rpm and from 5100 to 6000 with 100 rpm increment. Five consecutive periods of the measured variables were always stored. If the signal drop-out was recognized, the data were re-measured. The IRC ppr was set to 720 offering resolution of 0.5° of the cam. The measurement conditions are noted in Table 10.1 and Table 10.2. The measurements were not started before a 15-minute warm up procedure in which the oil temperature reached at least 40 °C.

Since the figures presented in the following text depict data characterizing real valvetrains that are on the market, we decided to withhold the real data scales of the kinematic variables. The *"1" normalized* values (data rescaled to range from -1 to 1) are shown instead. This approach does not compromise the validity of the data since the purpose is to carry out the comparison of the variables of the two valvetrains.

The data of the camshafts speed fluctuations presented in this chapter were smoothed as described in chapter *5.1.5 Camshaft speed fluctuations and the IRC accuracy*.

The convention is to denote the valves closest to the camshaft-crankshaft connection (in our case provided by a timing chain) as the 1st. In_1 will denote the first *intake* valve closest to the camshaft drive. Out_1 will denote the first *exhaust* valve closest to the camshaft drive – see Figure 10.1.

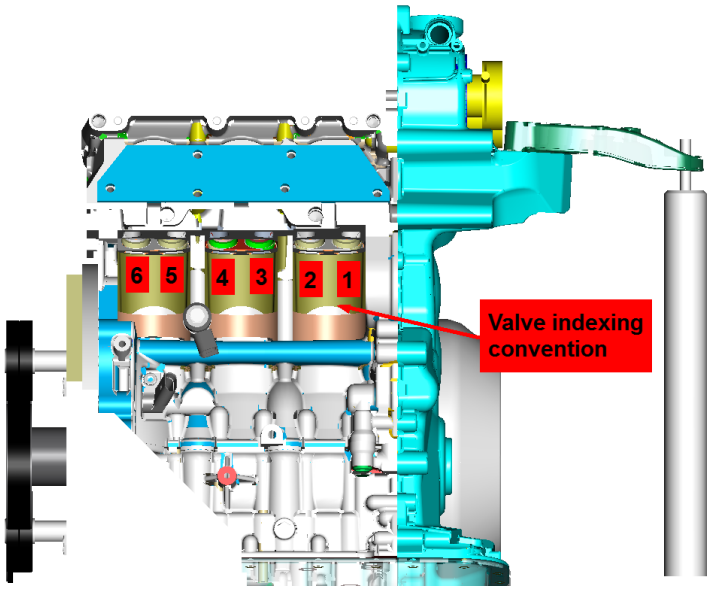


Figure 10.1 – Valve indexing convention

Table 10.1 – Measurement conditions, standard valvetrain

The ambient temperature [°C]	22 - 25
The oil pressure [bar]	4.2 (1000 rpm) – 5.2 (6000 rpm)
The oil temperature [°C]	40-55

Table 10.2 – Measurement conditions, low-friction valvetrain

The ambient temperature [°C]	22 - 25
The oil pressure [bar]	3.5 (1000 rpm) – 4.4 (6000 rpm)
The oil temperature [°C]	40-55

10.1 Measurement time duration

One of the main motivations for the development of this automated system was to reduce the duration of the measurement. The comparison of the measurement times can be found in Table 10.3. The times show the duration of the measurement procedure of one valve. It is important to point out that the old system did not allow for a delay of the IRC reference mark, which resulted in a physical repositioning of the IRC when moving from valve to valve (and thus additional time). This has been overcome thanks to the technique described in *6.2 Trigger pulse delay*.

The measurement procedure was always repeated 10 times, with the average time calculated. The time saving of the developed automated system is apparent. However, it is even more important that the application saves data that are not affected by the drop-out phenomenon and thus does not demand additional processing before performing the standard analyses.

Table 10.3 – Measurement duration results

	Average duration of the measurement procedure (15 data sets, 1000 rpm to 6000 rpm)
Original system manual slip correction, visual data testing, manual data saving	cca 10 min (reported by the test engineers of ŠKODA AUTO a.s)
Developed automated system slip correction ON: ± 5 rpm; data testing ON	2:25 min (in average 5 data files were re-measured because signal drop-out or other phenomenon was recognized)
Developed automated system slip correction OFF; data testing OFF	1:29 min

10.2 Measurement of the camshaft speed fluctuation

Measurement of the camshaft speed fluctuations can help to discover a faulty valvetrain component or, as it was expected, to improve the precision of calculations that presume the speed of the shaft to be constant. It also helps to identify speeds that are critical for the valvetrain. With the old measurement system it was not possible to carry out this type of measurement. Our apparatus together with the developed software performs the measurement of the camshaft speed fluctuations in parallel to the measurement of the valve kinematic variables. It should be noted that since the engine is not under fired condition the fluctuations

aren't caused by combustion. The pistons were removed and the original crankshaft was replaced with a modified straight 'dummy' shaft, only to drive the connected valvetrain and the oil pump. The rpm fluctuations are caused mostly by the valve spring forces, the shape of the cams and the resulting driving moment, chain vibrations and the resonance of those parts [33]. It is obvious that without combustion, the speed fluctuations are different than in the real engine. Nevertheless, this information is still important for the above-mentioned reasons.

Figure 10.2 shows a comparison of the speed fluctuations of the intake camshaft of the standard and the low-friction valvetrain at rpm_{crank} = 1000. Figure 10.3 depicts those values for the exhaust shaft of the two valvetrains. At this speed, the fluctuations of the low-friction valvetrain have slightly lower amplitude. The frequency of the camshaft sprocket-teeth can be clearly identified in both the data: thirty-six teeth.

Figure 10.4 and 10.5 show the speed fluctuations for rpm_{crank} = 5100. As it is shown in the next chapter, at this speed the deviation from the average speed is maximal. At this speed the amplitude of the fluctuations is nearly identical for both the standard and the low-friction valvetrain camshafts.

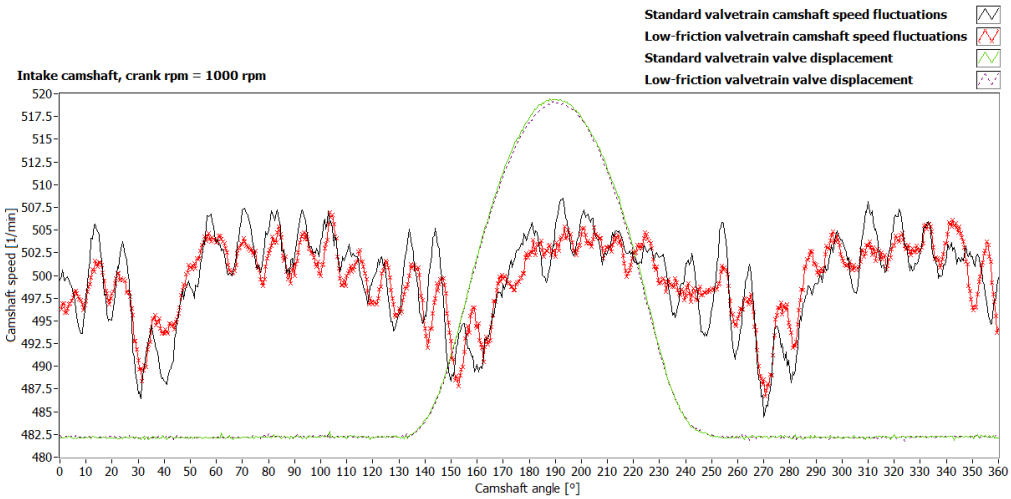


Figure 10.2 – Comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, intake camshaft, In_1, rpm_{crank} = 1000

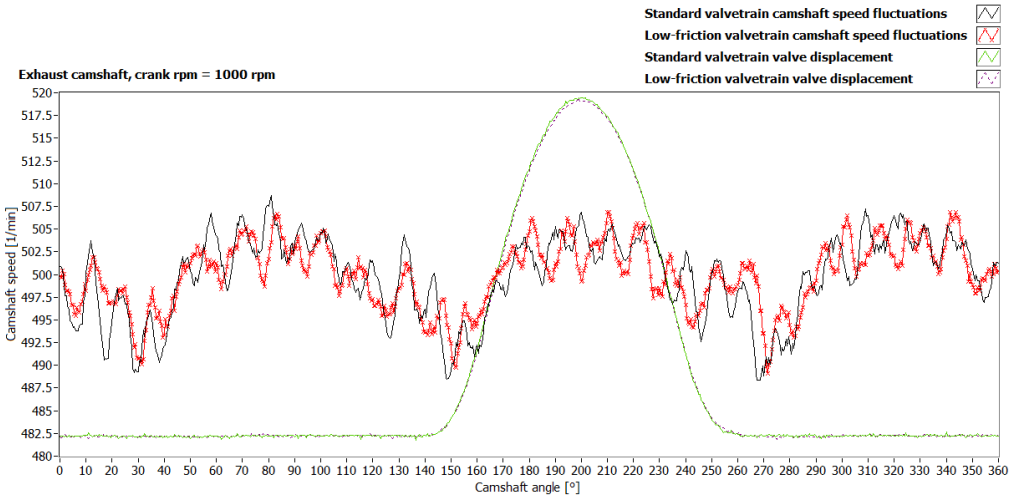


Figure 10.3 – Comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, exhaust camshaft, Out_1, rpm_{crank} = 1000

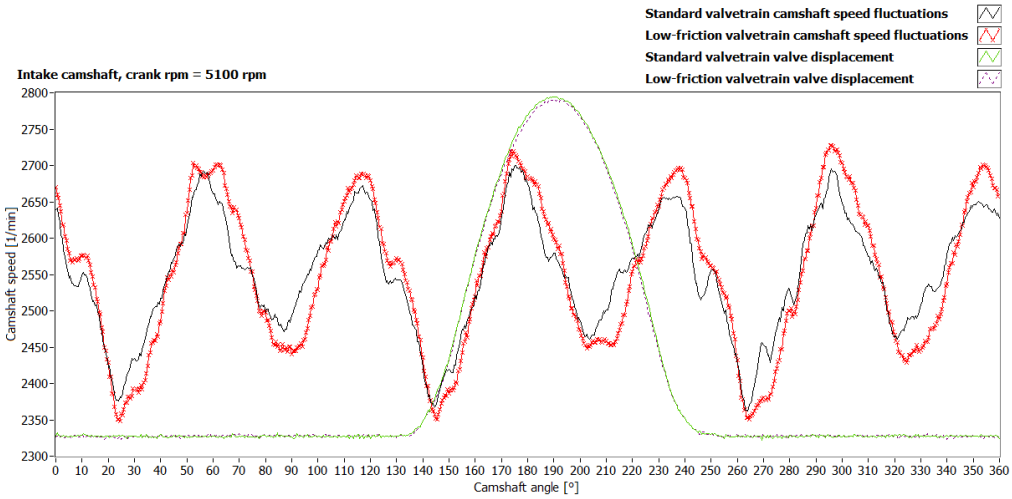


Figure 10.4 – Comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, intake camshaft, In_1, rpm_{crank} = 5100

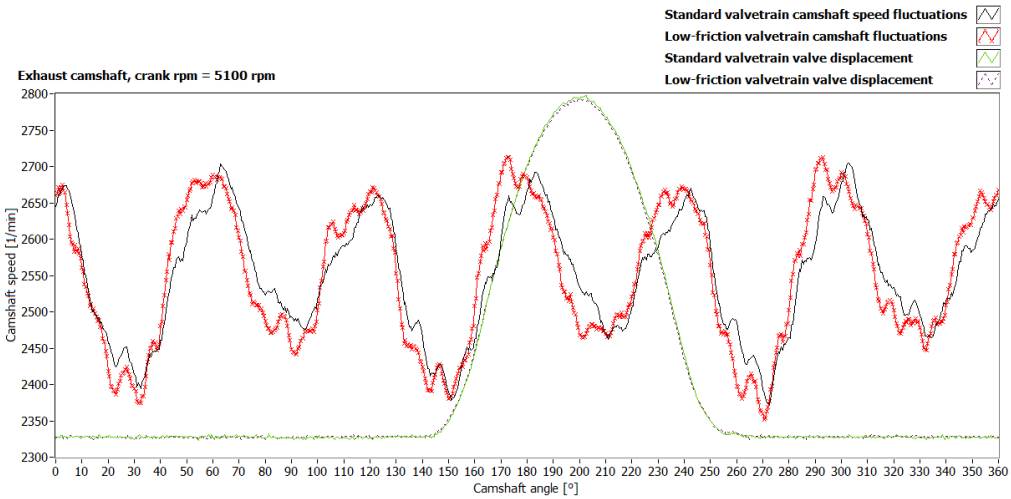


Figure 10.5 – Comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, exhaust camshaft, Out_1, rpm_{crank} = 5100

Figure 10.6 corresponds to Figures 10.2 and 10.4 and depicts the speed fluctuations in the frequency domain expressed as orders of the base (camshaft) frequency. At rpm_{crank} = 1000 the 3rd order is dominant (3-cylinder engine) being followed with its multiples: 6, 9, 12, 15… with decreasing tendency in the magnitude. The multiples of three are present since the resulting camshaft driving moment is not a precise sinusoidal (Figure 10.8), so the spectrum smears to following multiples of the base frequency. With higher rpm (see rpm_{crank} = 5100 in Figure 10.6) the situation changes and the 6th order (255 Hz) becomes dominant. This is caused by a change in shape of the resulting driving moment on the camshafts and in our case amplified by the resonance in the timing chain (explained in the following chapter). With high rpm the acceleration (and thus the inertia force) during opening and closing significantly increases, while the force of the springs remains the same. As a result, the shape of the torque

curve transferred by the camshafts changes (Figure 10.9). Figure 10.10 depicts the values of amplitudes of the significant orders in the resulting moment on a single camshaft with three cam lobes. The amplitude of the sixth order increases with increasing speed. Figure 10.11 then depicts the force between the cam lobe and the roller and demonstrates the origin of the increasing amplitude of the 6th order in the driving moment.

Figure 10.7 depicts the speed fluctuations of the exhaust camshaft in the frequency domain. The same conclusions as in case of Figure 10.6 can be drawn.

No significant differences between the standard and low-friction valvetrain were identified from the point of speed oscillation. No trace of malfunction was identified. The low-friction valvetrain has slightly lower amplitude of speed fluctuations at low speeds. With increasing speed the amplitudes become equal. Both valvetrains seem to perform as expected.

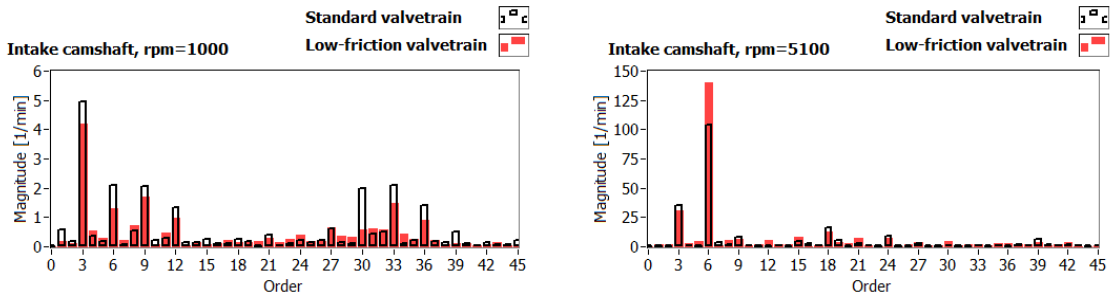


Figure 10.6 – Harmonic analysis, comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, *intake* camshaft

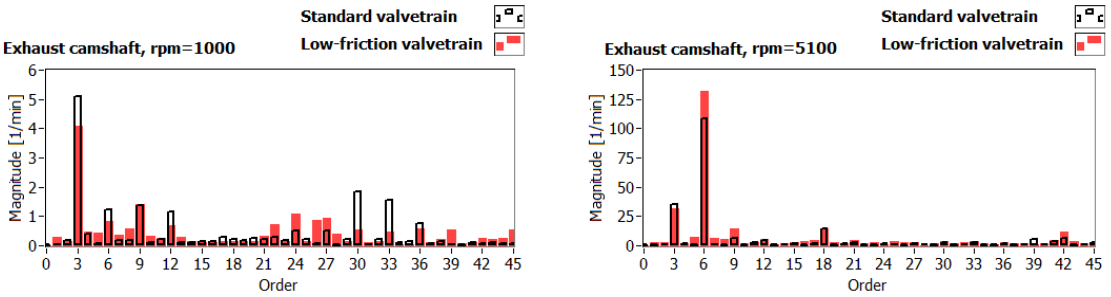


Figure 10.7 – Harmonic analysis, comparison of camshaft speed fluctuations of standard valvetrain and low-friction valvetrain, *exhaust* camshaft

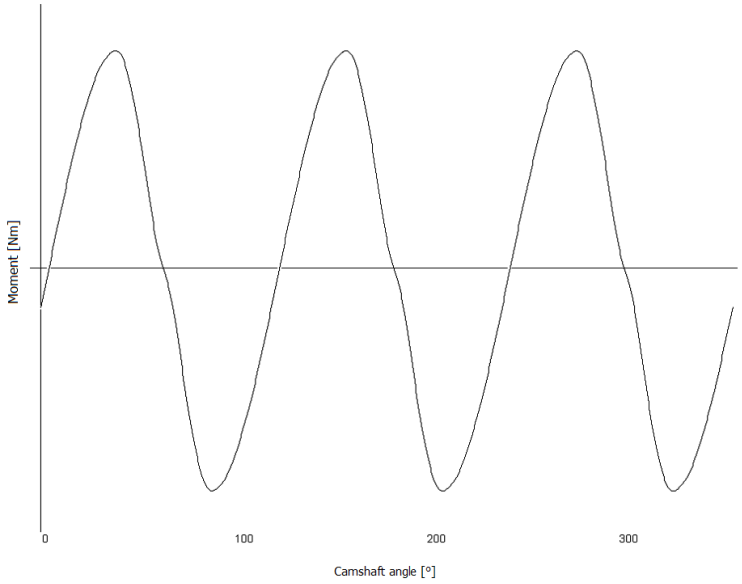


Figure 10.8 – Camshaft driving moment, low rpm (1000 rpm), simulation ŠKODA AUTO a.s.

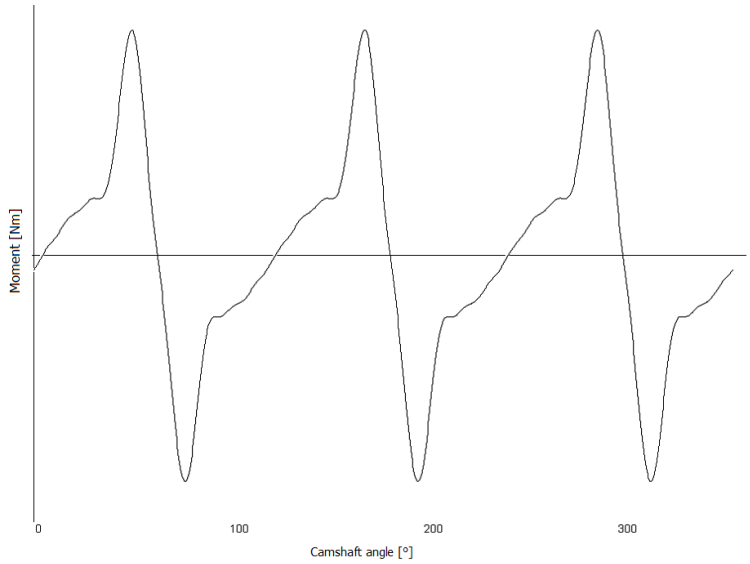


Figure 10.9 – Camshaft driving moment, high rpm (6000 rpm), simulation ŠKODA AUTO a.s.

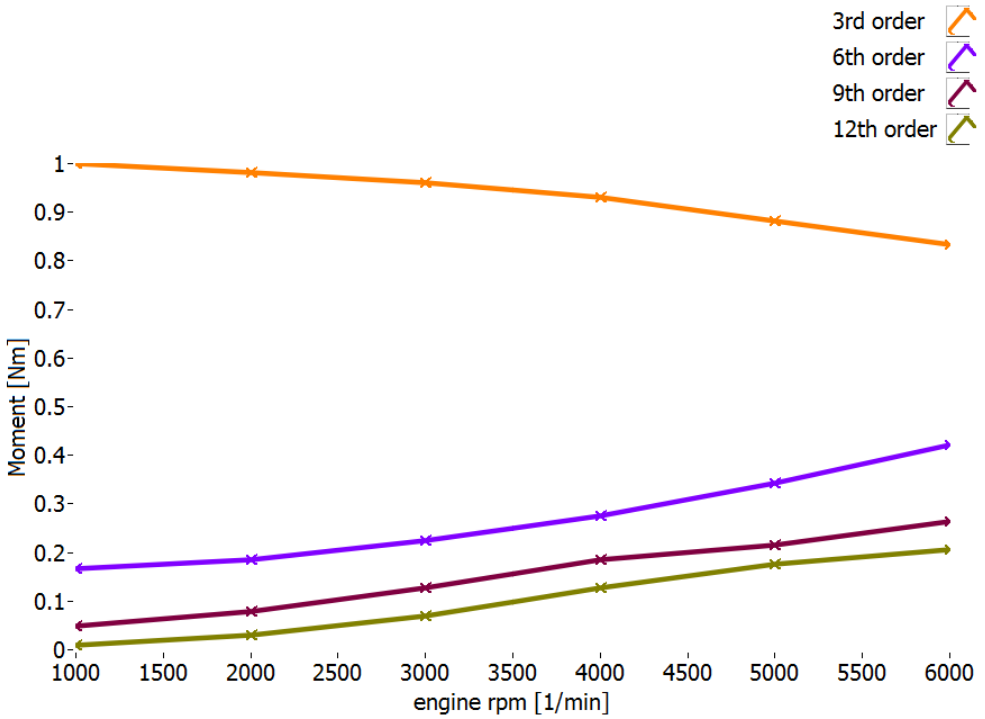


Figure 10.10 – Amplitude of dominant orders in the camshaft driving moment, simulation ŠKODA AUTO a.s.

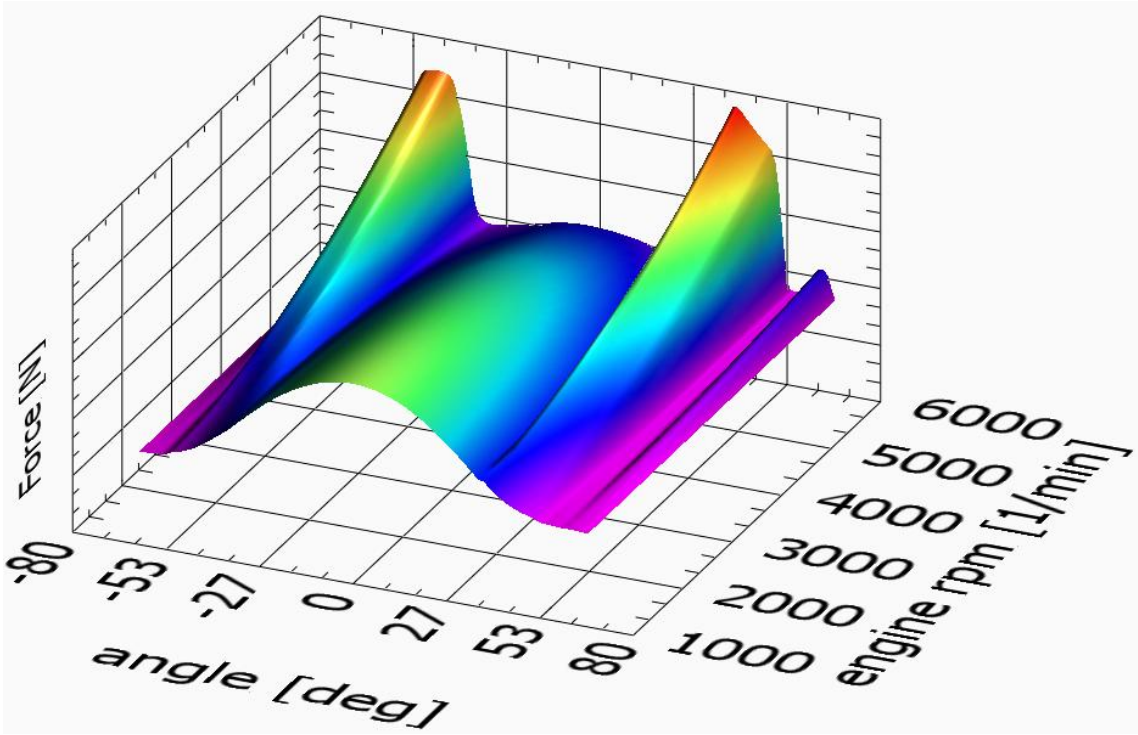


Figure 10.11 – Force between the cam lobe and the roller, simulation ŠKODA AUTO a.s.

10.3 Camshaft speed fluctuations across the whole engine speed spectrum

One of the beneficial analyses which can be carried out thanks to our complex system is the analysis of the camshaft speed fluctuations across the whole spectrum of the operational engine speeds. Information about the percentage deviation of the camshaft speed from its average value at a given engine (crankshaft) speed can be obtained. For this reason we carried out the measurement from 300 rpm to 6000 rpm with a 100 rpm step.

The following figures show the dependency of the maximal negative deviation from the average camshaft rpm (- *deviation*, red dashed-dotted line) on the engine rpm expresses as percentage of the actual speed. The green dotted line (+ *deviation*) denotes the maximal positive deviation. The black solid line is the mean of those two. Fig 10.12 shows those values for the intake and the exhaust camshafts of the standard 1.2 HTP engine valvetrain together with 3rd, 6th, 9th and 12th order of the base (camshaft) frequency. Fig 10.13 depicts the same for the low-friction valvetrain. In Fig 10.14, the mean fluctuation curves of both valvetrains are overlaid for easier comparison. The values from 300 to 500 rpm show high deviation which is not caused by the combustion engine but by the driving electromotor. At such a low speed the electromotor cannot ensure driving at a constant speed due to the variation of the torque during one revolution of the crankshaft. The inertia of the moving parts doesn't help to smooth the motion and allow the electromotor to revolve with a constant torque (thus a constant driving speed).

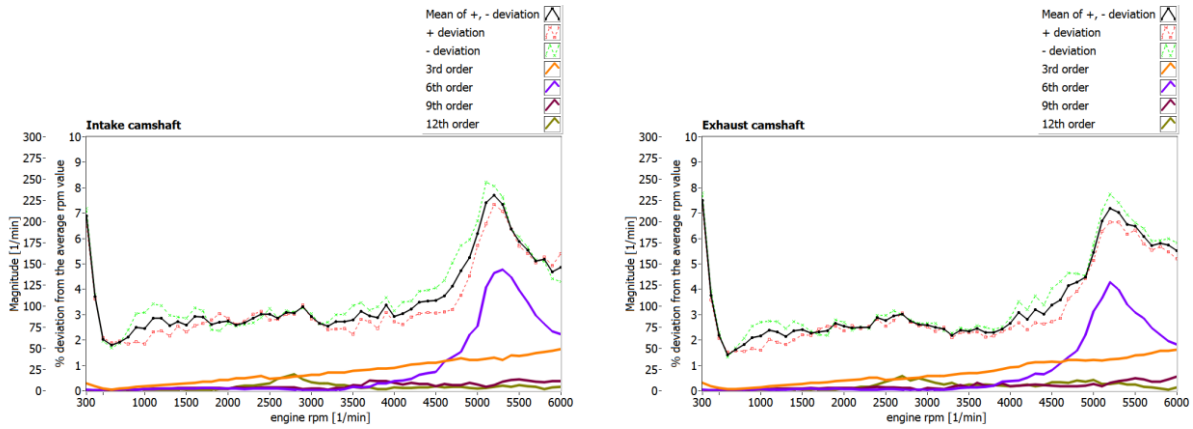


Figure 10.12 – Percentual deviation from the average rpm value and the orders of base camshaft frequency across the whole engine operational spectrum, *standard* 1.2 HTP valvetrain, half-engine setup

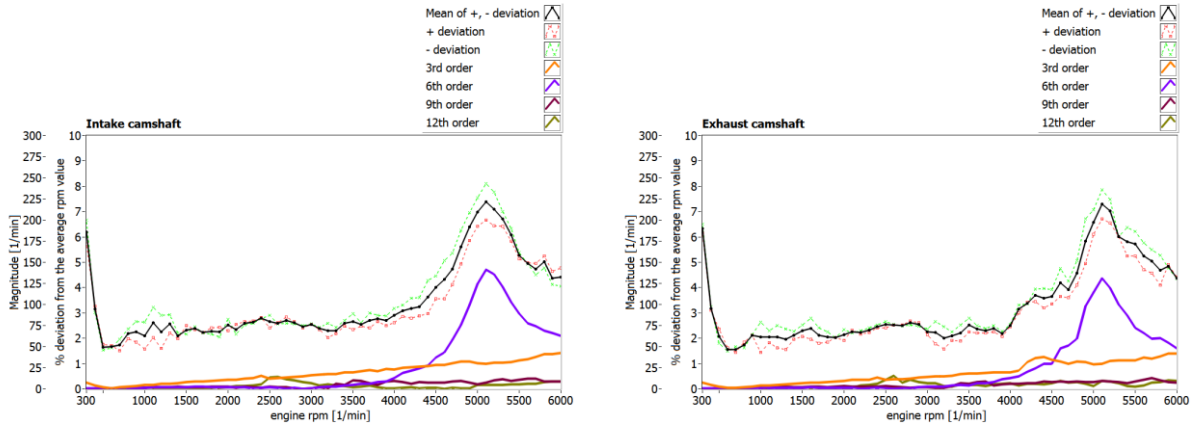


Figure 10.13 – Percentual deviation from the average rpm value and the orders of base camshaft frequency across the whole engine operational spectrum, *low-friction* 1.2 HTP valvetrain, half-engine setup

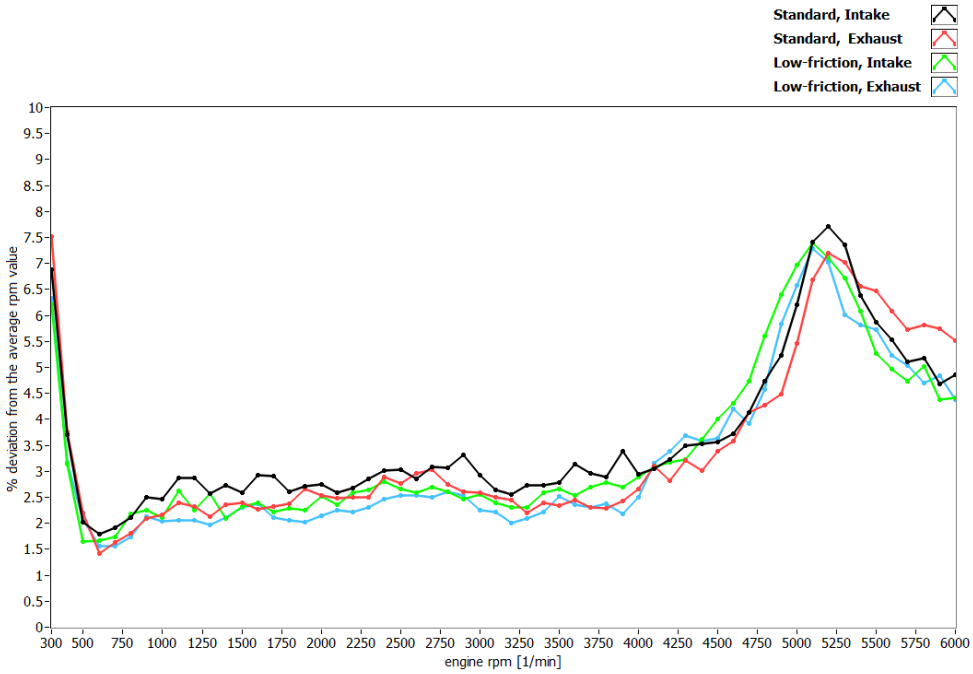


Figure 10.14 – Mean percentual deviation from the average rpm value across the whole engine operational spectrum, comparison of *standard* and *low-friction* valvetrain

The deviation has a clear rising trend with its peak around 5100 rpm. The peak is presumably caused by the resonance of the timing chain (resonance of 6th order; 255 Hz). The other possibility is the resonance of the chain tensioner blade. To discover the true origin, additional experiments would have to be carried out. The plastic tensioner blade could be exchanged by a more rigid type, and the fluctuations remeasured. As expected, after passing this critical frequency the deviations begin to drop. The same trend can be seen in both the intake and the exhaust camshaft data of both the standard and the low-friction valvetrain. Both of the valvetrains show approximately the same level of speed fluctuations across the measured speeds.

10.4 Measured kinematic variables

When designing the cam lobes and when verifying the resulting kinematic variables it is necessary to be able to compare the measured (valve displacement and velocity) and calculated (valve acceleration) curves across the whole spectrum of the measured speed. A specific type of normalization serves this purpose. Without this normalization it would be difficult to carry out a direct comparison of the data from different speeds (see Figure 10.15). If the time information is removed from the measured curves the velocity and acceleration will become invariant to the speed of rotation and will depend only on the angle. It can be expressed as:

$$\omega = \frac{d\varphi}{dt} \tag{10.1}$$

$$v = \frac{ds}{dt} \tag{10.2}$$

$$v_n = \frac{\frac{ds}{dt}}{\omega} = \frac{\frac{ds}{dt}}{\frac{d\varphi}{dt}} = \frac{ds}{d\varphi} \tag{10.3}$$

v_n is then the normalized velocity of the valve. The units are mm/rad. The same situation for the valve acceleration:

$$a = \frac{dv}{dt} = \frac{d^2s}{dt^2} \tag{10.4}$$

$$a_n = \frac{\frac{d^2s}{dt^2}}{\omega^2} = \frac{\frac{d^2s}{dt^2}}{\left(\frac{d\varphi}{dt}\right)^2} = \frac{d^2s}{d\varphi^2} \tag{10.5}$$

a_n is the normalized acceleration of the valve. The units are mm/rad^2.

It is important to emphasize that the figures in this chapter (Figures 10.16, 10.17, 10.19 and 10.20) plot data that were subjected to two types of normalization with different objectives. The first normalization removes time dependency from the data, enabling direct comparison. The following *"1" normalization* rescales the data in -1 to 1 range since we are not permitted to publish the real data scales. This double normalization might be a little confusing.

Since the information about the actual shaft speed fluctuations is not usually available it is a well-adopted standard to use the average of the angular frequency and neglect the camshaft speed fluctuations. Since our apparatus acquires the speed fluctuations in parallel to the kinematic variables, we can verify whether neglecting of the speed oscillations is a reasonable simplification. The comparison is depicted in Figure 10.16. The normalization let us compare the data from rpm_{crank} = 1000 and 5100. The red and green lines were calculated while using the speed fluctuations data. It is apparent that neglecting this information does not introduce a significant error with respect to the shape of the curves. On the other hand, at 5100 rpm, where the highest deviation from the average of the speed was encountered (see chapter *10.3 Camshaft speed fluctuations across the whole engine speed* spectrum), the differences in the values of the extremes (the first and the second maximum, resp. minimum of the velocity) are already noticeable. The same conclusion can be drawn for the acceleration normalization (Figure 10.17).

As a result, the speed fluctuations do not have to be taken into account as long as they are not of high significance. If the deviations were reaching 20% or more, it would be beneficial to include the speed fluctuation information into the calculation.

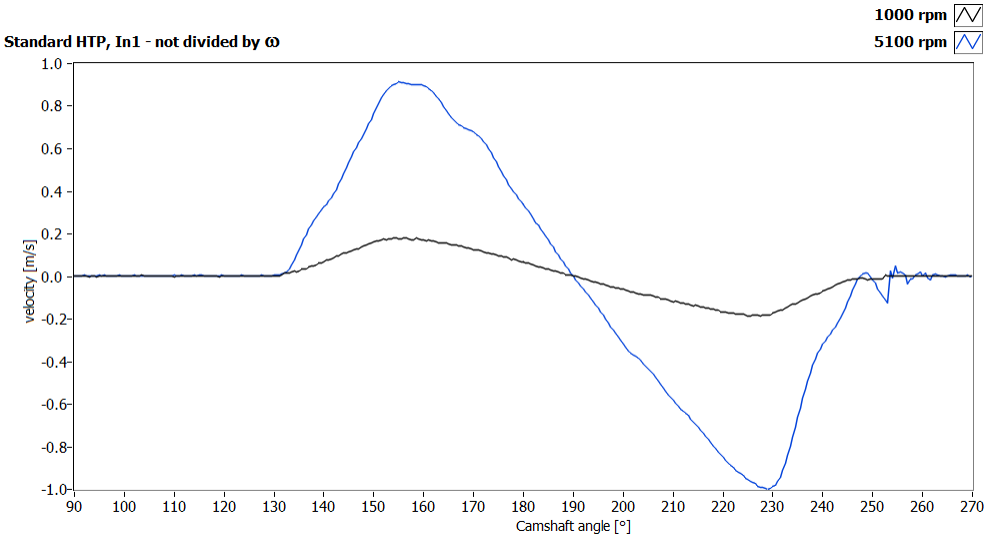


Figure 10.15 – Valve velocity, *standard* 1.2 HTP valvetrain, In_1

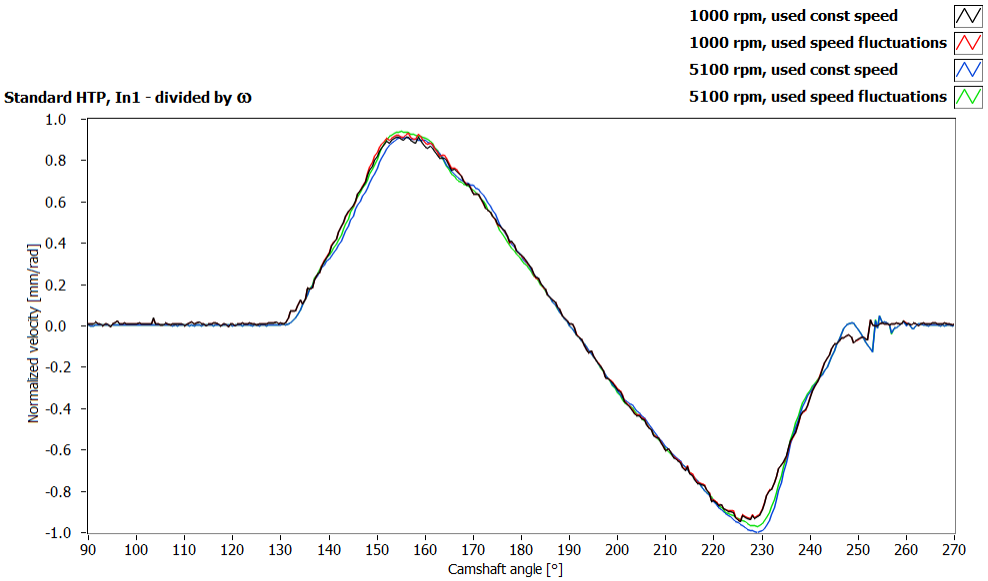


Figure 10.16 – Calculated normalized valve velocity, *standard* 1.2 HTP valvetrain, In_1

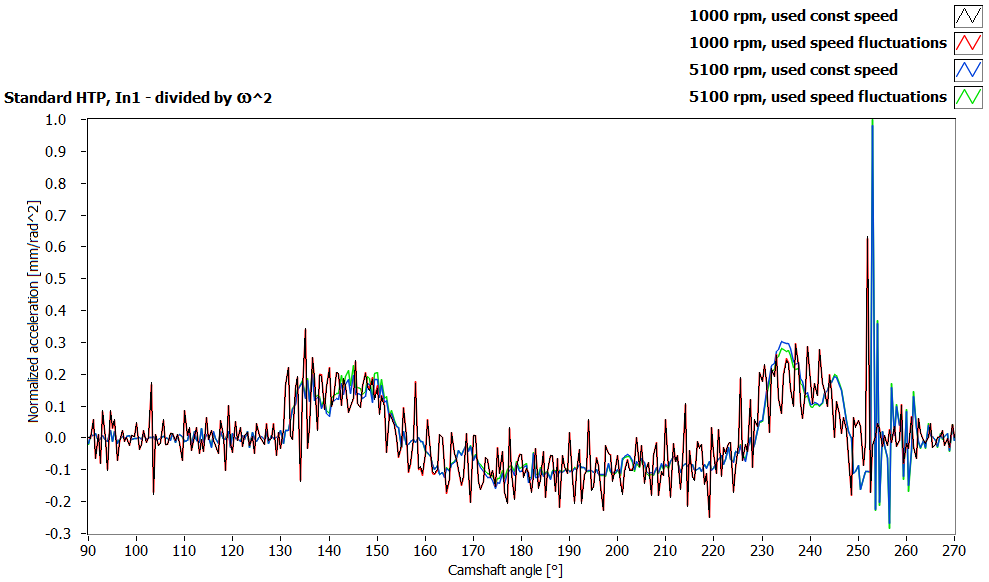


Figure 10.17 – Calculated normalized valve acceleration, *standard* 1.2 HTP valvetrain, In_1

The following figures depict the normalized kinematic variables of the standard and the low-friction valvetrain measured at different speeds.

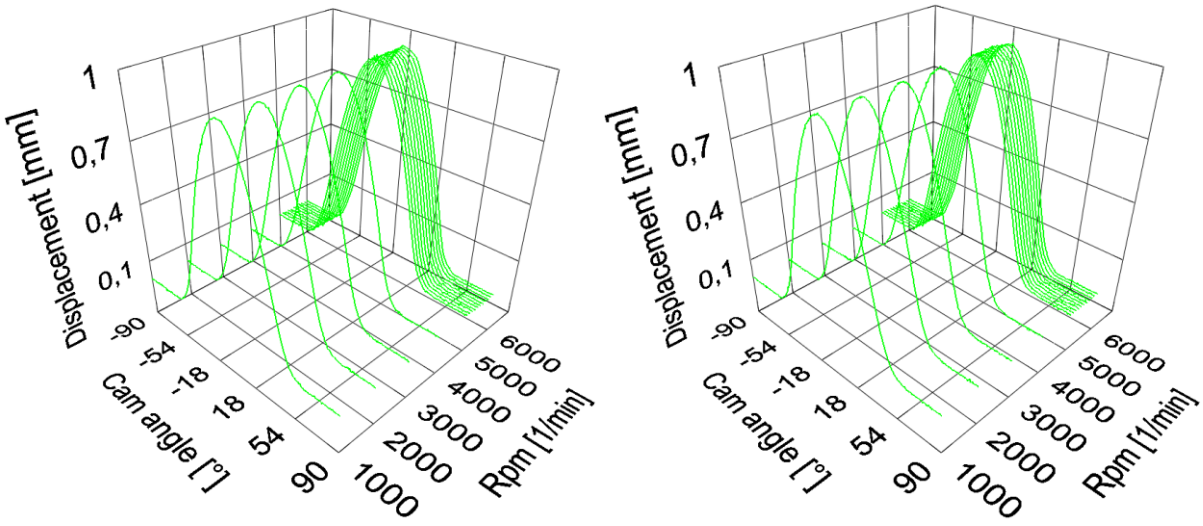


Figure 10.18 – Valve displacement, *standard* valvetrain (left) vs. *low-friction* valvetrain (right), In_1

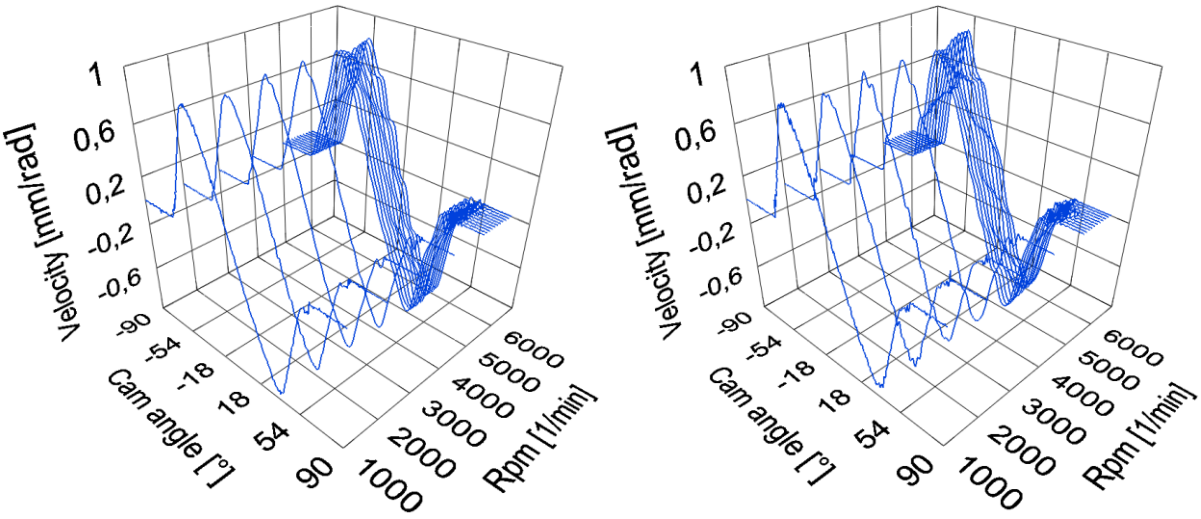


Figure 10.19 – Normalized valve velocity, *standard* valvetrain (left) vs. *low-friction* valvetrain (right), In_1

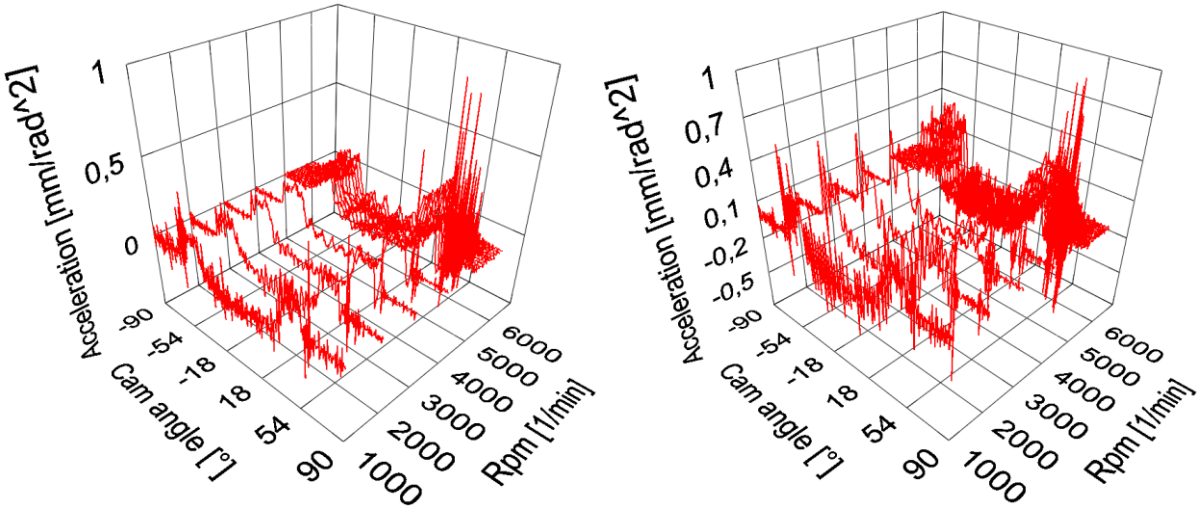


Figure 10.20 – Normalized valve acceleration, *standard* valvetrain (left) vs. *low-friction* valvetrain (right), In_1, forward difference

Only the data of the first intake valve are shown. Other valves, whether intake or exhaust, show a similar correlation between standard and low-friction valvetrains. Therefore, they are not presented.

10.5 Computation of valve acceleration using the speed fluctuation information

Valve displacement and velocity are measured. However, from the point of dynamics and verification of the proper function of a valvetrain, the most valuable source of information is acceleration. The most common approach is to obtain it from the measured velocity using a numerical derivative. The reason is that it is complicated to obtain acceleration of a valve

directly, accurately, and reliably at a high rpm. A piezoelectric accelerometer can be difficult to attach to the valve and its mass influences the resulting data. Special low-weight (1 gram) accelerometers can be purchased but even then the problems with connecting cables that tend to separate at high rpm persist. Nevertheless, there were some attempts made usually at low speeds and with the accelerometers attached to various parts of the valvetrain components [51], [52], [53], [54]. Since the high speed laser Doppler vibrometers became the standard device for measuring the kinematics of the valves, acceleration has been obtained indirectly from the velocity [6].

In order to obtain acceleration, we usually utilize either a forward or backward numerical derivative. The reason is that these simple methods reveal step changes in the velocity signal (for example, the first contact of the valve with the seating [7]). The advanced methods tend to smooth the data so these phenomena would remain unnoticed.

It is important to mention that the measured signal is sampled with respect to the cam angle, and not the time. Thus, the time derivative is not an option. If we want to obtain acceleration, we have to use the following approach:

$$a = \frac{dv}{dt} \quad \text{if we use the equation (10.1) we get} \quad a = \frac{dv}{d\varphi} \cdot \omega \tag{10.6}$$

Using the numerical differences it can be written as

$$a = \frac{\Delta v}{\Delta \varphi} \cdot \omega \tag{10.7}$$

If we neglect the accuracy of the IRC the angle difference will be constant. Expressed in radians:

$$\Delta\varphi[rad] = \Delta\varphi[deg] \cdot \frac{\pi}{180}; \qquad \Delta\varphi[deg] = \frac{360}{ppr}; \quad \rightarrow \quad \Delta\varphi[rad] = \frac{2\pi}{ppr}; \tag{10.8}$$

The camshaft angular velocity is

$$\omega = 2\pi f_{cam} = \frac{2\pi\, rpm_{cam}}{60} \tag{10.9}$$

Using the equations (10.8) and (10.9) in (10.7) we have

$$a = \frac{\Delta v\, rpm_{cam}\, ppr}{60} \tag{10.10}$$

Common practice is to neglect the variance in rpm_{cam} and presume the rotational speed to be constant. The main reason is that synchronized information about camshaft speed fluctuations is not usually available. Let us consider the measured data in Fig. 10.21 where the valve velocity is synchronized with the camshaft speed (the acquisitions started with the same trigger). The speed of 5100 rpm was chosen because the highest deviation from the average of the speed was encountered at this speed (see chapter *10.3 Camshaft speed fluctuations across the whole engine speed* spectrum).

In Fig 10.22 we can see the comparison of the acceleration calculated while presuming a constant camshaft speed (black) and the approach that gets the benefit of the measured speed fluctuations (red). The formula (10.10) and the forward finite difference were used.

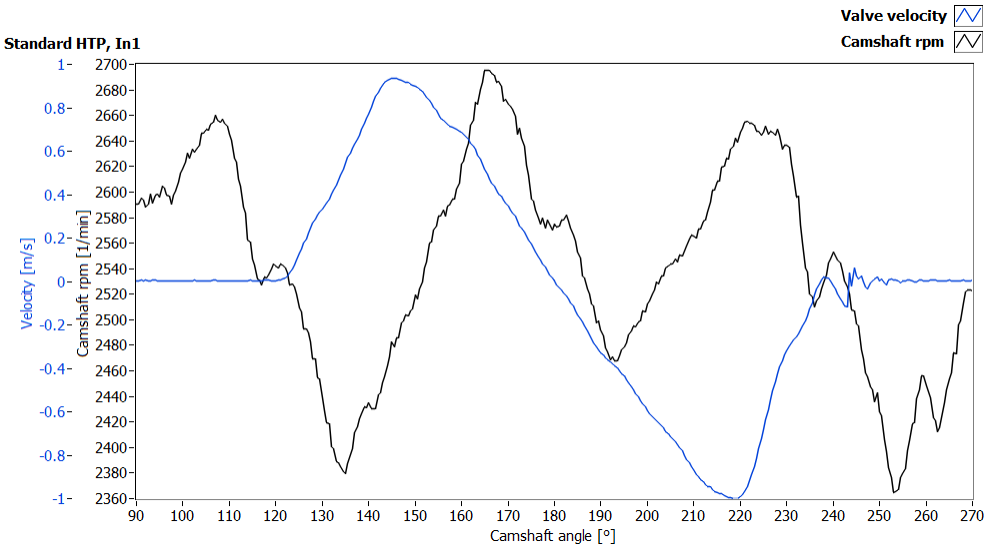


Figure 10.21 – Valve velocity and camshaft speed, *standard* 1.2 HTP valvetrain, In_1, rpm_{crank} = 5100

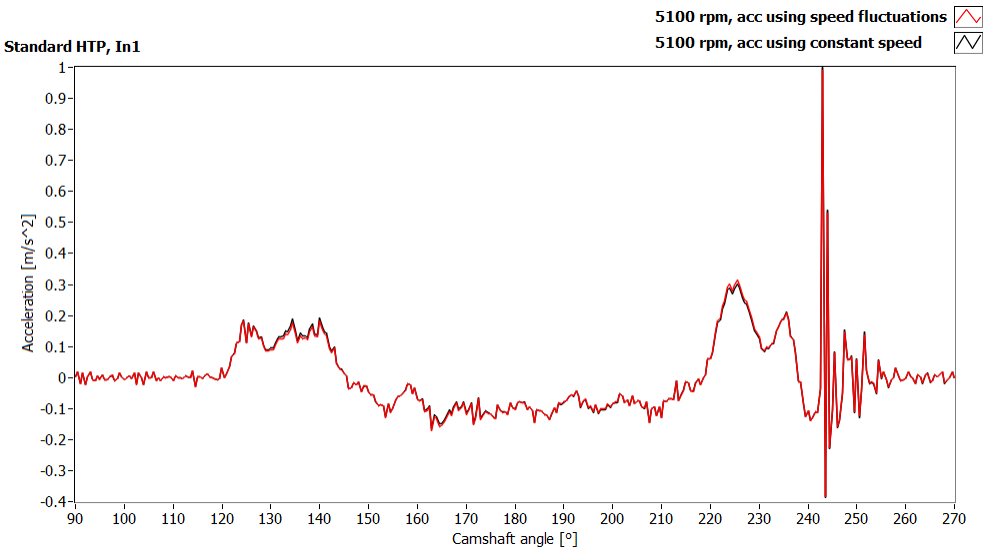


Figure 10.22 – Calculated acceleration of the valve, rpm_{crank} = 5100, In_1, forward difference, half-engine setup

Only slight differences between the calculated curves can be observed. It is safe to say that in the case of modern valvetrains, rotational irregularities do not play a significant role in the valve acceleration calculation and the rotation can be simplified as constant for this purpose. Even in situations when the speed fluctuates by 15% as in the case of a partial-engine setup at 6000 rpm (see chapter *11*), the differences would be still insignificant. See Figure 10.23.

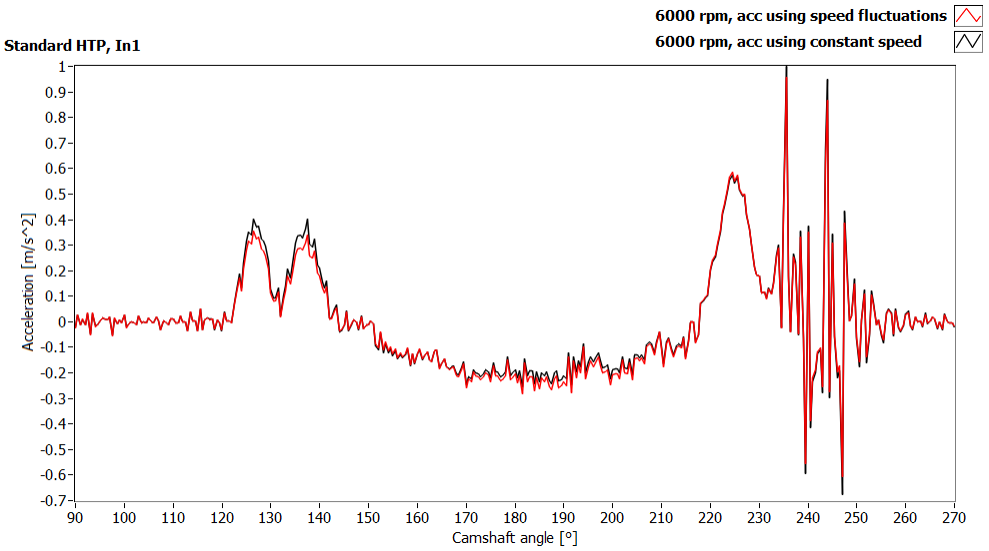


Figure 10.23 – Calculated acceleration of the valve, rpm_{crank} = 6000, In_1, forward difference, partial-engine setup

10.6 Valve float and deviations from ideal kinematic motion

One of the phenomena that are usually tested is the valve float. Valve float can occur at a high rpm when the inertia forces of the valvetrain components exceed the spring force holding the valve on the cam profile, allowing the components to separate [5]. This reduces engine efficiency, performance and lifetime and potentially increases the engine emissions. If the cam lobe loses contact with the roller (or the bucket tappet), or if the valve loses contact with the rocker arm, it will lead to an increment of the maximal valve lift. This is undesired and dangerous especially for the valvetrains with hydraulic lash adjusters. It should never appear during operating speeds of the serial production valvetrain. If the valvetrain separated, the hydraulic lash adjuster would compensate the created clearance. This, as a result, would leave the valve slightly open during the compression. In the worst case, contact of the valve with the piston could occur.

The data of the valve displacement at 1000 rpm were taken as a reference. At this low speed the valve precisely follows the profile of the cam lobe. In Figure 10.24, comparison of the reference valve displacement (solid line) and the high rpm (6000 rpm, dashed line) valve displacement is depicted for the standard (left) and low-friction (right) valvetrain. In case of the standard valvetrain, it is tempting to say that the valve undergoes valve float since the valve lift increment is recognizable (middle red region) and was measured by $\Delta s_{max} = 0.05$ mm. Nevertheless, it is not a result of the separation of the valvetrain components. At low speeds during the maximum of the lift, i.e. between the inflection points of the valve displacement (the domain of the negative valve acceleration), the valvetrain is tightly connected by the force of the spring. At high speeds, great inertia forces of the moving parts act against the spring force on the mentioned domain thus resulting in lower contact force. However, the valvetrain does not separate. Figure 10.25 depicts the acceleration of the valve at 6000 rpm. If the valvetrain really separated, we would see a clear peak between 200° and 220° where it would reconnect.

Besides the increment of valve displacement we also see deviations from the reference curve during valve opening and closing. This is caused by a combination of three main factors. The first is the delay in closing the check ball in the hydraulic lash adjuster, which becomes noticeable at high speeds. It results in slower opening at 6000 rpm (first red region). Since the hydraulic lash adjuster gains a lower position during opening, it will also close faster (third red region). If this were the only influence, it would create a constant offset from the reference trajectory.

The second contributing factor is bubble formation in oil, which is more likely to occur with high rpm. The air bubble, unlike the oil, is compressible and will allow the hydraulic lash adjuster to gain a lower position for a short period of time.

The third contributing factor is a slight bending of the rocker arm.

The same discussed effects can be observed during the opening and closing phase of the low-friction valvetrain; however, no increase of the maximal valve lift was measured. The described differences are also present in the measured data of the remaining valves. No valve float was detected and from this point the valvetrains seem to perform adequately.

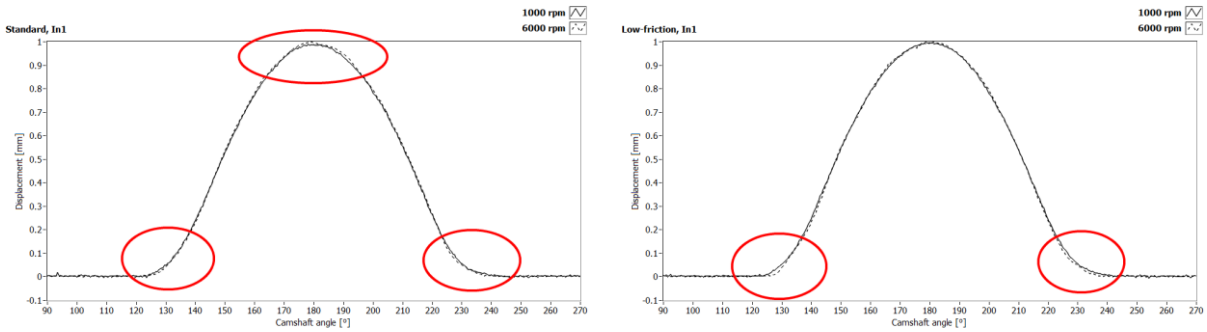


Figure 10.24 – Comparison of valve displacement at 1000 rpm and 6000 rpm, standard valvetrain (left), low-friction valvetrain (right), In_1

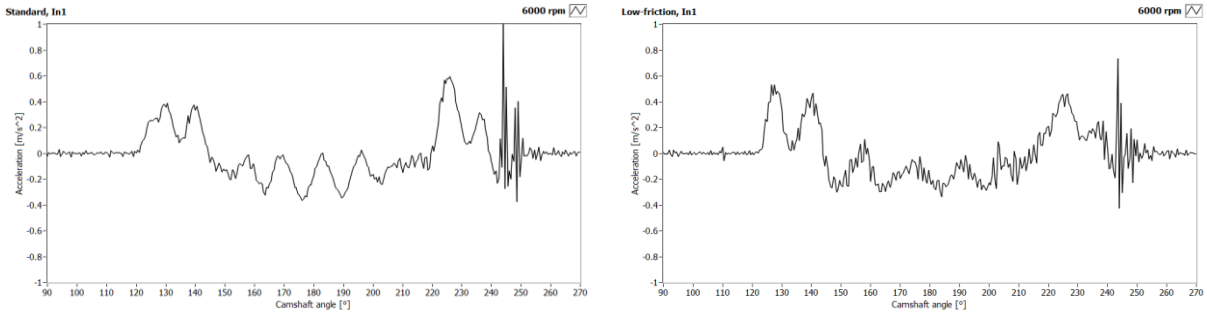


Figure 10.25 – Calculated acceleration at 6000 rpm, standard valvetrain (left), low-friction valvetrain (right), In_1, forward derivative

10.7 Valve bouncing and impact velocity

One of the goals of the valvetrain design is to reduce both valve float and valve bounce. Valve bouncing can occur if the valve closes with velocity which is too high – for example, if weak springs are used which cause the valvetrain to separate and the valve to bounce a few times before fully closing. As can be seen from Figures 10.18-10.20 and 10.24 no valve bouncing was detected.

The impact velocity is the velocity with which the valve is seated. The exact value is determined from the valve kinematics data by localizing dominant peak in the acceleration during the valve closing process. This is very difficult and demands experience and knowledge of what can be happening during the valve closing. The impact velocity was evaluated from 4000 rpm to 6000 rpm (Tables 10.4 and 10.5), since lower speeds are not interesting for this analysis. Between 1000 rpm and 2000 rpm the impact velocities are very low and the exact point of impact is usually unidentifiable. Only the data from the closest and endmost valve were evaluated. These valves are influenced the most by the torsional vibrations and the bending of the camshaft. All five recorded consecutive periods of data from each speed were examined in order to reduce the risk of not discovering a faulty valvetrain. In some cases it was not possible to identify the impact velocity. In other cases only an interval could be determined but not the exact value.

Since we are not permitted to publish the absolute values of the impact velocities the presented values were rescaled and should be interpreted as the percentage of the maximal impact velocity encountered in this thesis.

No unusually high impact velocities were detected. The valvetrain is considered functional if the impact velocity is less than 1 m/s (1000 mm/s) – the empirical parameter. All the impact velocities fulfill this criterion. The values of the low-friction valvetrain are lower by approximately 20% than those of the standard valvetrain, for the most part (see Figure 10.26).

Table 10.4 – Impact velocities of standard valvetrain

	Impact velocity [% $v_{max\ impact}$]	***4000 rpm***	***5000 rpm***	***5100 rpm***	***5200 rpm***	***5300 rpm***	***5400 rpm***	***5500 rpm***	***5600 rpm***	***5700 rpm***	***5800 rpm***	***5900 rpm***	***6000 rpm***
In_1	Half-engine setup	39 36 38 38 39	75 72 70 57 74	68 74 75 78 70	64 59 58 53 59	72 68 74 71 72	45 72 64 63 62	32 62 68 62 66	64 76 71 72 72	54 32 46 43 36	46 39 43 51 39	51 61 51 36 55	42 46 51 50 47
In_6	Half-engine setup	36 39 33 29 39	59 55 46 57 61	39 34 50 49 62	61 83 53 51 68	67 62 59 61 61	62 57 61 67 61	63 59 57 62 32	43 42 36 61 42	53 36 39 38 46	42 45 46 39 49	21 45 38 47 36	43 58 47 46 54
Out_1	Half-engine setup	- - - 28 -	62 70 66 59 67	43 45 30 36 61	68 71 37 57 53	74 16 71 67 32	76 79 38 38 49	83 32 78 33 61	49 59 41 45 45	59 36 54 58 63	- 59 30 34 29	47 54 46 47 21	30 30 29 37 43
Out_6	Half-engine setup	30 28 32 28 29	- 54 - - -	41 - - - -	- - - - - *(v<39)*	38 - 53 - 53	- 61 50 34 55	34 - 41 32 43	- 51 58 42 -	- 43 - 42 37	32 32 37 28 53	32 28 - 30 28	- - - - - *(v<39)*

Table 10.5 – Impact velocities of low-friction valvetrain

	Impact velocity [% $v_{max\ impact}$]	*4000 rpm*	*5000 rpm*	*5100 rpm*	*5200 rpm*	*5300 rpm*	*5400 rpm*	*5500 rpm*	*5600 rpm*	*5700 rpm*	*5800 rpm*	*5900 rpm*	*6000 rpm*
In_1	Half-engine setup	28 24 30 14 8	43 20 - 22 25	29 20 34 28 42	17 47 46 47 24	46 - 76 - -	- 57 47 28 55	37 54 45 51 -	- 42 - 32 -	14 - 13 36 20	- 32 - 45 45	21 12 34 18 34	13 11 16 24 26
In_6	Half-engine setup	45 34 39 36 34	37 53 37 34 51	28 41 30 33 34	- 24 50 43 -	46 - - 20 39	32 - 61 33 -	- 47 34 46 -	- 39 - - -	- - - 34 -	- - - - - (v<92)	- 59 - 63 -	Not measured
Out_1	Half-engine setup	39 34 - - -	- - - 43 50	- 39 47 - -	28 - - 30 -	38 41 41 61 46	- 46 - - -	- 38 32 - 54	22 26 47 38 33	39 36 50 25 54	63 54 37 42 49	24 33 41 37 47	55 - 33 22 39
Out_6	Half-engine setup	26 17 26 18 26	18 - 18 - -	39 - 42 - -	- - - - - (v<68)	- - 34 - - (v<100)	- - - 36 32	18 21 - 22 -	- - - - - (v<65)	- - - 51 54 42	- - 29 43 51	13 32 55 - 37	17 34 - 13 - (v<61)

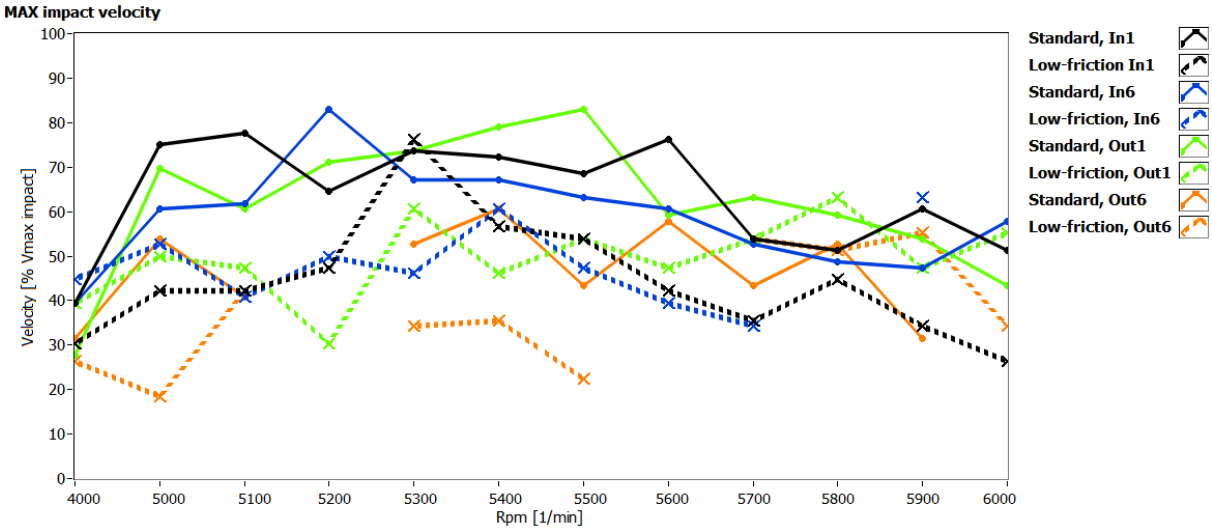


Figure 10.26 – Comparison of the maximal impact velocities of the *standard* (solid lines) and the *low-friction* (dashed lines) valvetrain

10.8 Summary of the measurements of the standard and the low-friction valvetrain

We measured a standard 1.2 HTP valvetrain in comparison with a low-friction valvetrain for the same engine. This measurement demonstrated the overall capabilities of the designed and constructed apparatus. The algorithm for signal drop-out detection performed well and was capable of discovering this unwanted phenomena and repeating the measurement.

Camshaft speed fluctuations were successfully measured in synchrony with the kinematic variables of the valves. It provided additional information and helped discover the resonance in the valvetrains. It was also expected that using the speed fluctuations for the velocity and acceleration normalization and the valve acceleration computation would improve the

precision of the results. Nevertheless, the differences in the results that used the speed fluctuations and the results that considered the speed to be constant were insignificant. It is safe to say that in the case of modern valvetrains that do not experience huge camshaft speed fluctuations, this can be neglected for the above-mentioned calculations.

Based on the analyses carried out, the valvetrains seem to operate as expected. The low-friction valvetrain performs as good as the standard valvetrain. Thanks to its design it operates with lower friction losses and thus requires less input power. An engine with such a valvetrain will have lower fuel consumption and will produce less emissions. The next step in the process of verification of the proper function of the valvetrains would be a lifetime testing.

11 Comparison of measured values of valvetrain in half-engine setup and partial-engine setup

In industry, the measurement of kinematic variables of OHC valvetrains is usually simplified. Rather than testing the complete engine, assembly of merely the head and head cover is used. We will call it *partial-engine setup* in the following text. Such a setup is easy to assemble, and does not demand additional modifications of the components while, at the same time the exchange of the valvetrain components is fast and the valves are easily optically accessible.

The *half-engine setup*, in contrast, uses nearly a complete engine. It requires the pistons to be removed and the crankshaft replaced by a straight dummy shaft. Apertures have to be milled in place where the pistons belong in order to allow access of the laser Doppler probes to the valves. The timing chain cover has to be modified to allow attachment of the IRC sensor. The engine is attached to a rigid test frame and driven by an electromotor. The half-engine setup is as close as possible to the real engine assembly which is mounted in an automobile. For more details of the half-engine setup see description in *3.1 Developed measurement system – the hardware.*

Since preparation of the half-engine setup is time-consuming, the simplified setup is preferred. Our partial-engine setup (see Figure 11.1) consists of a rigid steel plate holding the apparatus. Two supports are attached to the plate, upon which the cylinder head and the cover are mounted. A head gasket separates the cylinder head from the supports and prevents oil leaks. One support has a channel for pressurized oil distribution since the engine's oil pump and distribution channels are not present (normally the oil is brought through the engine block). The engine oil pump is replaced by an external oil unit and distribution system. The oil from the head is collected through the oil outlets of the cylinder head. A custom side cover was manufactured to replace the missing timing chain cover, and a pulley is mounted at the end of the driven camshaft. The torque transfer between the driving electromotor and the pulley is assured by a ribbed belt. The IRC sensor is attached to the pulley to offer precise information about the position and speed of the camshaft.

Such a setup poses only a fraction of the original mass of the engine. Nevertheless, it is expected to deliver results close to the half-engine setup and thus close to the values of a real operating engine [20]. Although the partial setup has become a widely-used solution since the OHC valvetrains became standard, no direct comparison of the same valvetrain in half-engine and partial-engine setup has been published (to the author's knowledge). We conducted this comparison on a 3-cylinder ŠKODA 1.2 HTP engine – first with a standard valvetrain and then with its low-friction version. The main focus was on the valve kinematic variables, but the two setups were also compared from the point of camshaft speed fluctuations. The measurement conditions are noted in Tables 11.1 and 11.2. The IRC ppr was set to 720 offering resolution of 0.5° of the cam.

Since the figures presented in the following text depict data which characterize real valvetrains that are on the market, we decided to withhold the real data scales of the kinematic variables. The normalized (*"1" normalization* – normalized values in range from -1 to 1) values are shown instead. This approach does not compromise the validity of the data since the purpose is to carry out a comparison of the variables.

The data of the camshaft speed fluctuations presented in this chapter were smoothed as described in chapter *5.1.5 Camshaft speed fluctuations and the IRC accuracy*.

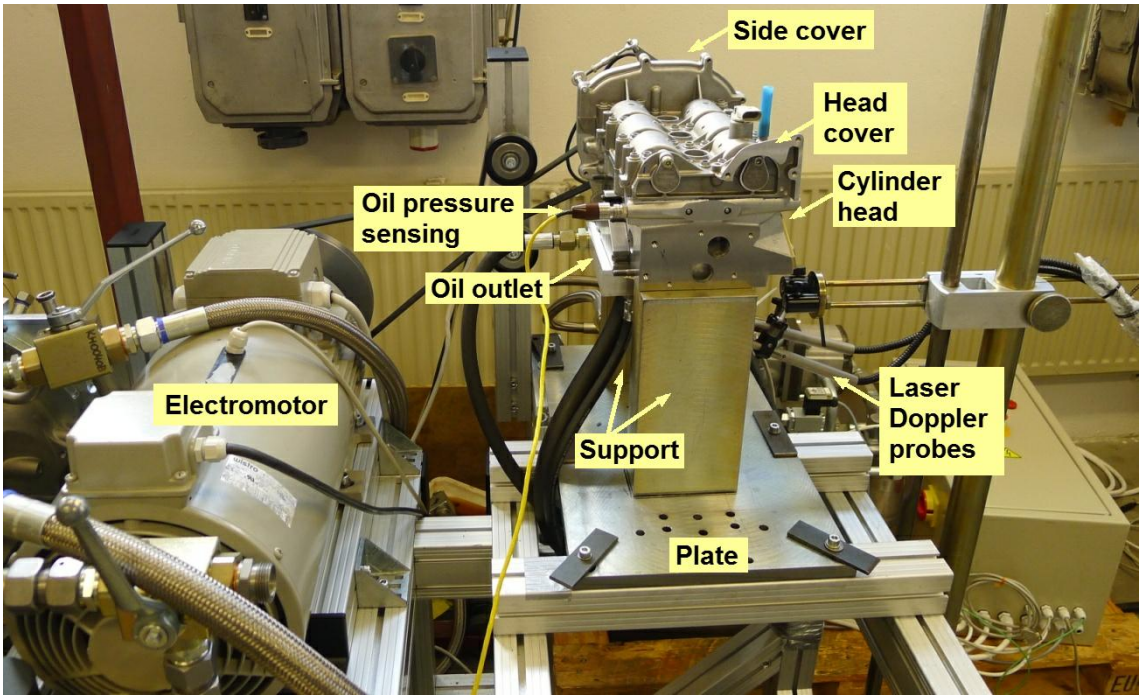


Figure 11.1 – Engine in partial setup

Table 11.1 – Measurement conditions, standard valvetrain

The ambient temperature [°C]	22 - 25
The oil pressure [bar]	4.5
The oil temperature [°C]	45-50

Table 11.2 – Measurement conditions, low-friction valvetrain

The ambient temperature [°C]	22 - 25
The oil pressure [bar]	3.5
The oil temperature [°C]	45-50

11.1 Camshaft speed fluctuations across the whole engine speed spectrum

Speed fluctuations of the partial-engine setup were expected to be significantly different from the half-engine setup. This is a result of missing moving parts and driving of only one camshaft at a time. The camshaft speed fluctuations of the partial-engine setup are caused mainly by the shape of the cams of the driven camshaft and the springs' force. The stiffness, length, tension and alignment of the belt that is used to drive the camshaft also play a significant role.

The following figures show the dependency of the maximal negative deviation from the average camshaft rpm (- *deviation*, red dashed-dotted line) on the engine rpm expresses as percentage of the actual speed. The green dotted line (+ *deviation*) denotes the maximal positive deviation. The black solid line is the mean of those two. Fig 11.2 shows those values for the intake and the exhaust camshafts of the standard 1.2 HTP engine valvetrain in the partial setup together with 3^{rd}, 6^{th}, 9^{th} and 12^{th} order of the base (camshaft) frequency.

Fig 11.3 depicts the same for the low-friction valvetrain. The same analysis for the half-engine setup can be seen in Fig 10.12 and Fig 10.13. The corresponding discussion can be found in chapter *10.3 Camshaft speed fluctuations across the whole engine speed spectrum.* Direct comparison is depicted in Fig 11.4 and 11.5.

The measurement was carried out from rpm_{crank} = 1000 rpm with 100 rpm increment. Uniform rotation of the driving electromotor was not achievable at lower speeds (< 1000 rpm) due to the variation of the torque during one revolution of the camshaft.

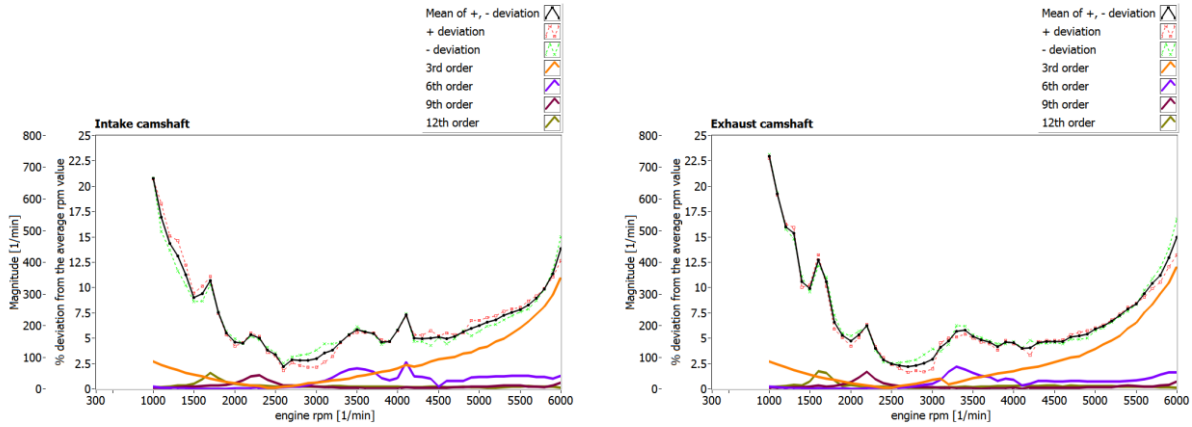


Figure 11.2 – Percentual deviation from the average rpm value and the orders of base camshaft frequency across the whole engine operational spectrum, *standard* 1.2 HTP valvetrain, partial-engine setup

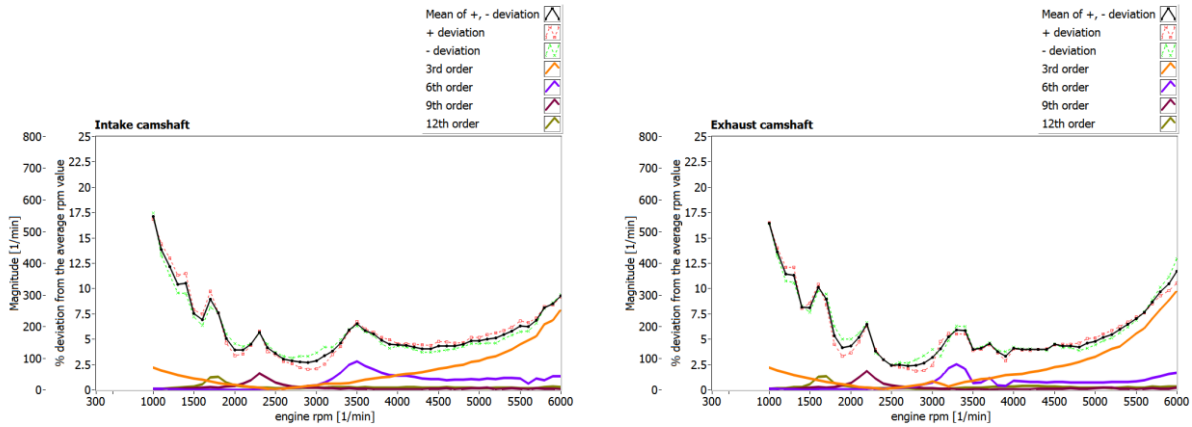


Figure 11.3 – Percentual deviation from the average rpm value and the orders of base camshaft frequency across the whole engine operational spectrum, *low-friction* 1.2 HTP valvetrain, partial-engine setup

At first the deviation has decreasing tendency up to 2500 rpm. With increasing speed the inertia forces increase and help to smooth the motion of the camshaft. Local maxima can be identified in the data. They are caused by the resonances of the strands of the belt (belt connecting the electromotor and the camshaft). Some of them were even clearly observed during the measurement, e.g. local maximum around 3500 rpm (resonance of 6th order). The belt resonance primarily occurs when the natural frequency of some length of the belt is excited by a frequency generated by the drive.

From 4500 rpm the percentual deviation begins to steeply rise again. The exact origin has not been identified. It would be interesting to carry out the measurement over 6000 rpm to discover where the resonance occurs, if it indeed does. Another option would be to repeat the measurement with a shorter and more rigid belt and observe the change in the trend. This would reveal if the resonance of the belt is involved. The last possible (but unprobable) explanation of the increasing speed fluctuation would be increasing torsional vibrations in the camshaft. To verify this theory, we would have to alter the moment of inertia of the camshaft; for example by adding metal rings and repeating the measurement.

The low-friction valvetrain shows lower deviations.

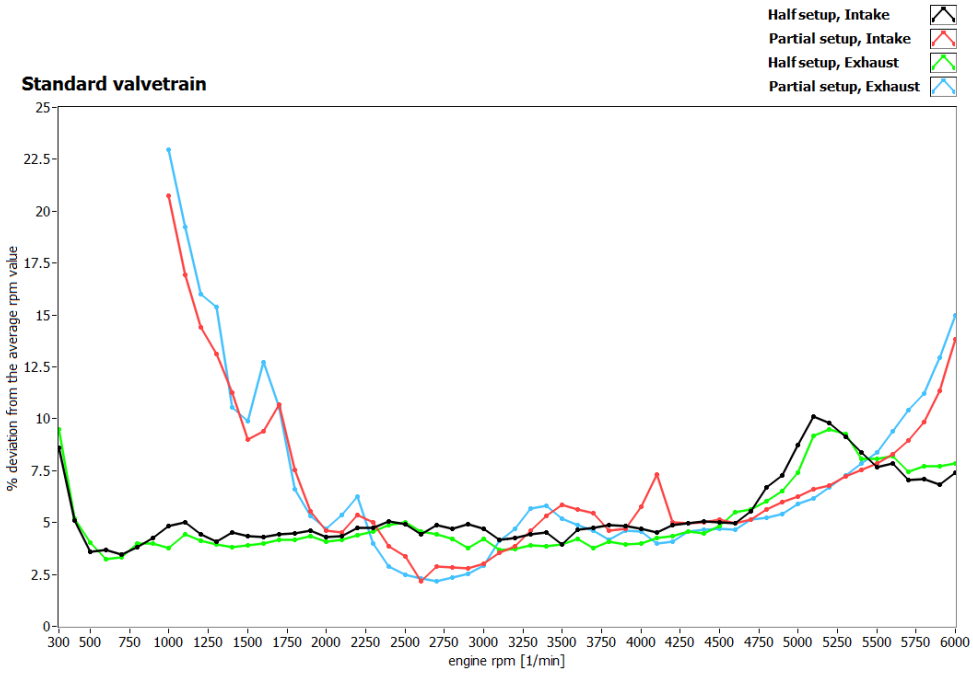


Figure 11.4 – Mean percentual deviation from the average rpm value across the whole engine operational spectrum, *standard* valvetrain, comparison of partial and half-engine setup

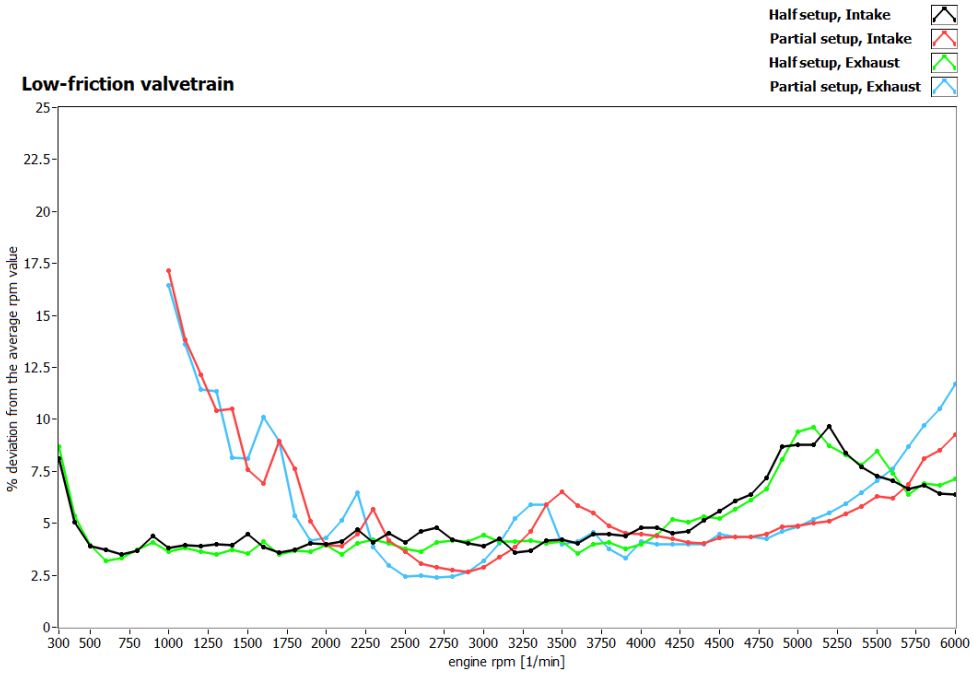


Figure 11.5 – Mean percentual deviation from the average rpm value across the whole engine operational spectrum, *low-friction* valvetrain, comparison of partial and half-engine setup

It can be concluded that the camshaft speed fluctuations significantly differ between the half-engine and the partial-engine setup at different speeds. The profound difference up to 2000 rpm is caused by the absence of the crankshaft and the second camshaft in the partial-engine setup. The inertia of the shafts and the counter torque of the cam lobes of the second camshaft help to smooth the rotation of the half-engine setup at a low speed. The differences with increasing speed are the result of resonances of different parts of the systems.

Nevertheless, these results were expected based on the differences of the two setups. It would be interesting to compare the measured data with the measurements of the partial-engine setup with both sprockets driven, contributing with the rotational irregularity of both camshafts. Such a system would be closer to the half-engine setup but would demand additional changes.

11.2 Direct comparison of the speed fluctuations

The differences in the speed fluctuations between the half-engine and the partial engine setup seem to be the key for understanding the differences in the measured valve kinematic variables. Figures 11.6-11.9 demonstrate in detail what was said in the previous chapter. The figures depict the speed fluctuations at 1000 rpm, 5100 rpm and 6000 rpm (where they differ) and also at 3000 rpm (where they are close to matching).

It is noticeable that the maxima of velocity of the valve (V_{MAX}, V_{MIN} = maximum of velocity during valve closing) differ between the half and the partial-engine setup. If we compare the speed fluctuations at these points, it is apparent (and not surprising) that a higher speed of rotation of the camshaft results in a higher valve velocity maximum. That is to say, the speed fluctuations directly influence the kinematic variables of the valve and since there are differences between the speed fluctuations of the half and the partial-engine setup, we can expect differences in the valve kinematics.

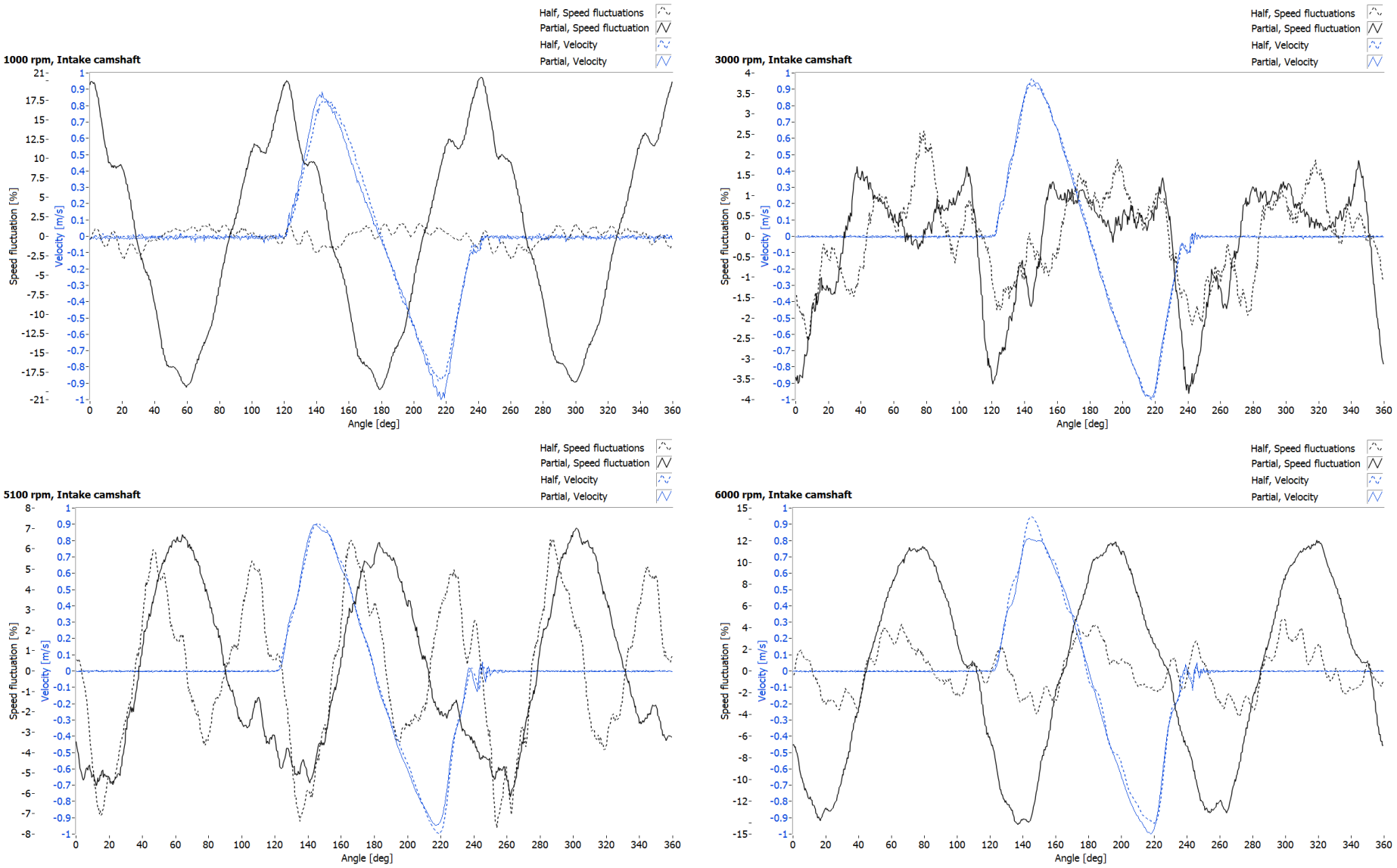


Figure 11.6 – *Standard* valvetrain, *Intake* camshaft, Comparison of speed fluctuations of half-engine setup (dashed lines) and partial-engine setup (solid lines)

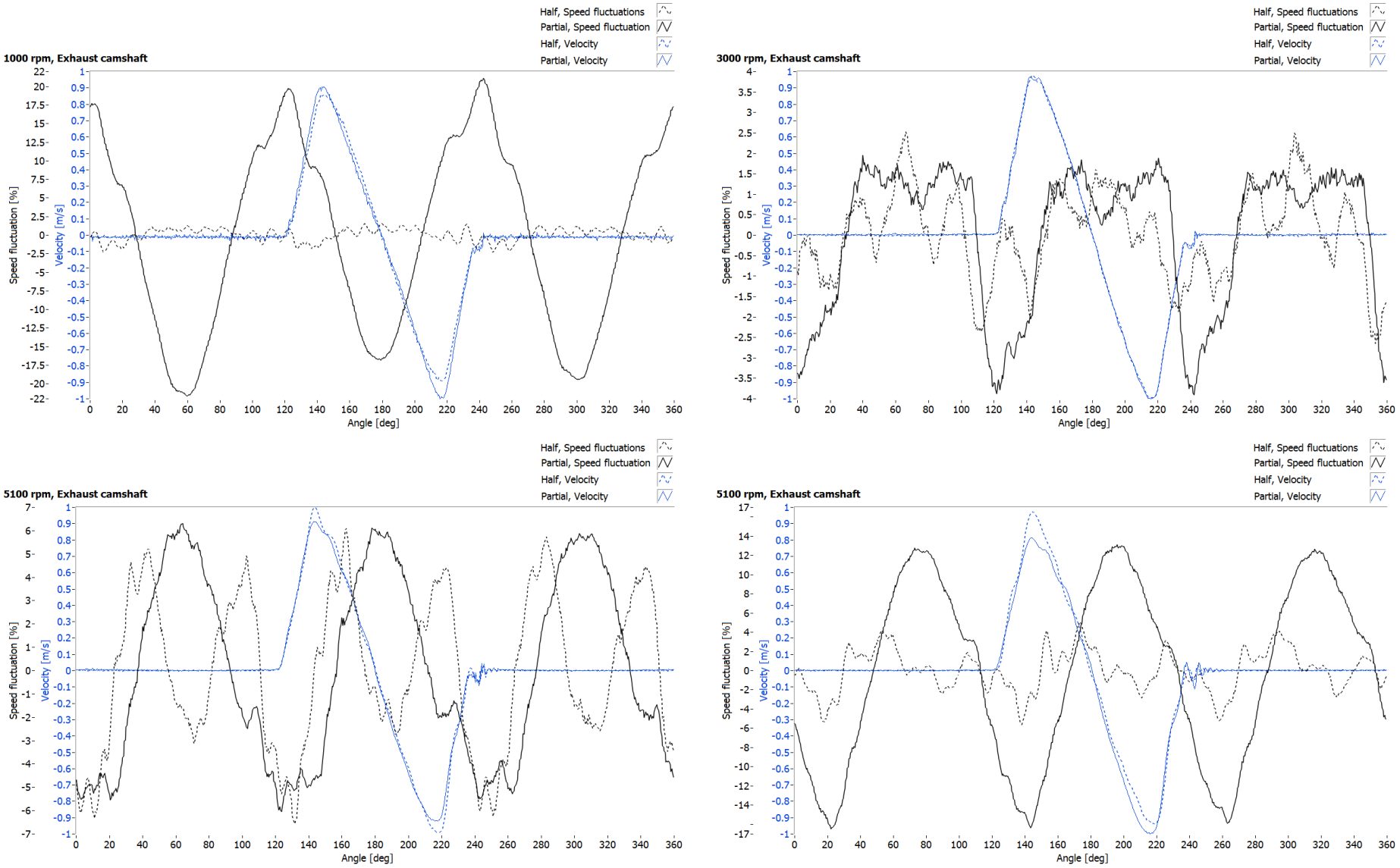


Figure 11.7 – *Standard* valvetrain, *Exhaust* camshaft, Comparison of speed fluctuations of half-engine setup (dashed lines) and partial-engine setup (solid lines)

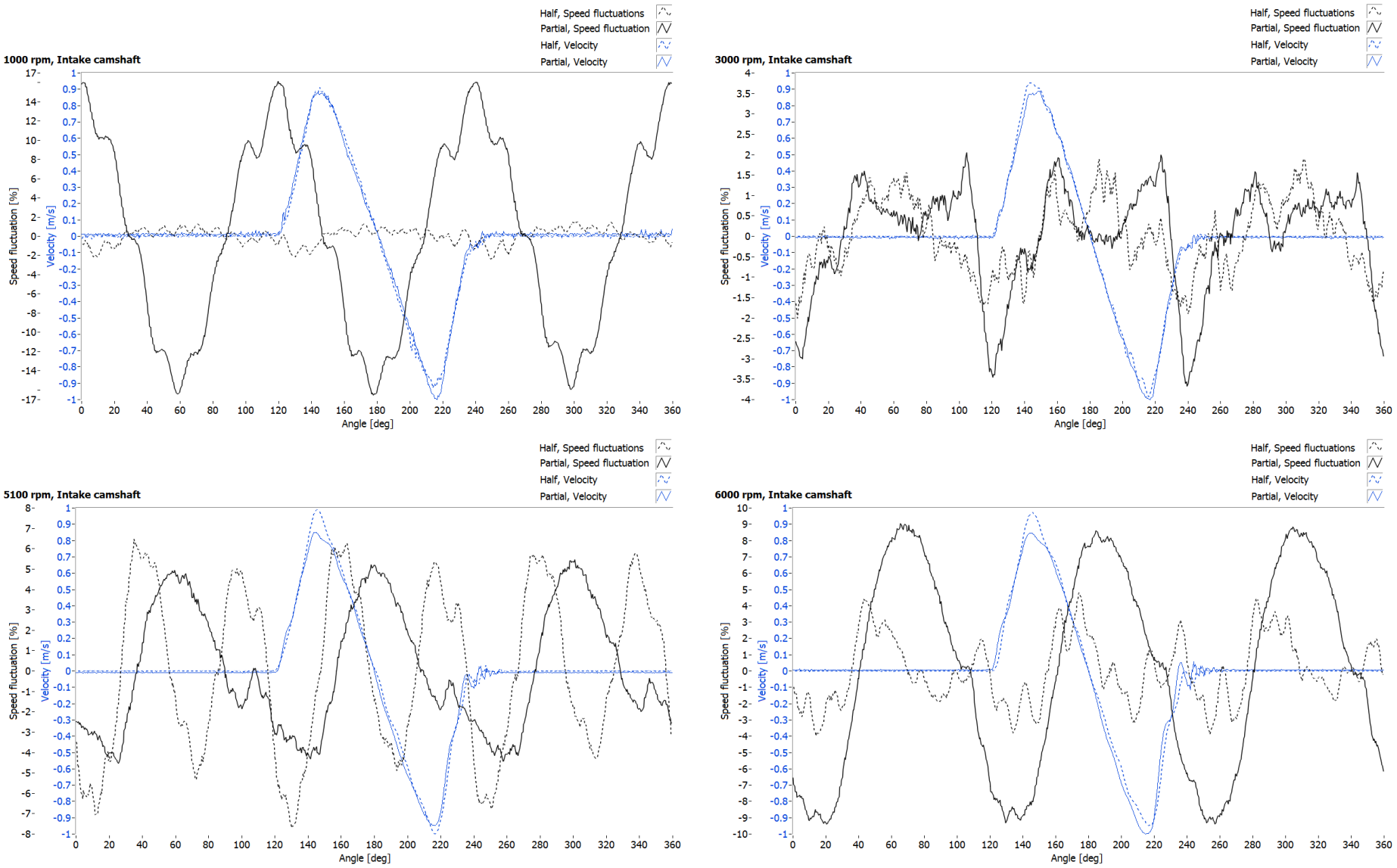


Figure 11.8 – *Low-friction* valvetrain, *Intake* camshaft, Comparison of speed fluctuations of half-engine setup (dashed lines) and partial-engine setup (solid lines)

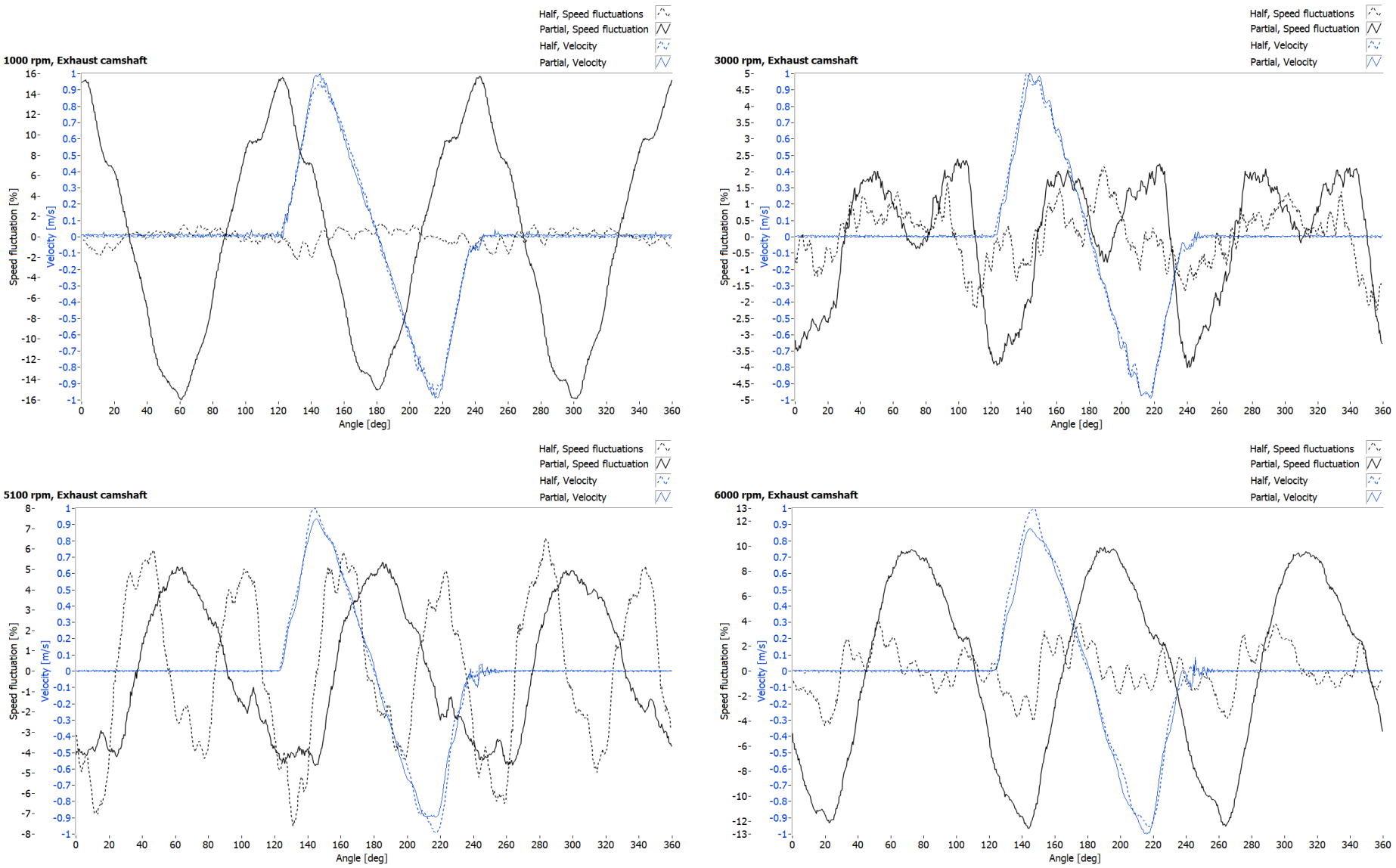


Figure 11.9 – *Low-friction* valvetrain, *Exhaust* camshaft, Comparison of speed fluctuations of half-engine setup (dashed lines) and partial-engine setup (solid lines)

The following figures (Fig 11.10-11.13) depict the speed fluctuations and how they change with the speed of the engine. The frequency of the camshaft sprocket-teeth (36 teeth) can be identified in the measured data of the half-engine setup at 1000 rpm (see also Fig. 10.2 and 10.3). With increasing speed (starting at 4500 rpm) the 6th order of the camshaft base frequency becomes dominant in the data of the half-engine setup (see also Figures 10.12 and 10.13 for detail). In the partial-engine data the magnitude of the 3rd order becomes dominant at 4000 rpm (see also Figures 11.2 and 11.3 for detail). Also a phase shift of the oscillations with increasing frequency can be recognized. This is clearly visible in the following figures, especially in the data of the partial-engine setup between 1000 rpm and 2500 rpm.

To identify the exact origin of the differences in the speed fluctuations (amplitude, phase, orders) between the two setups we would need precise numerical models. To construct them is beyond the scope of this thesis and leaves door open for future research.

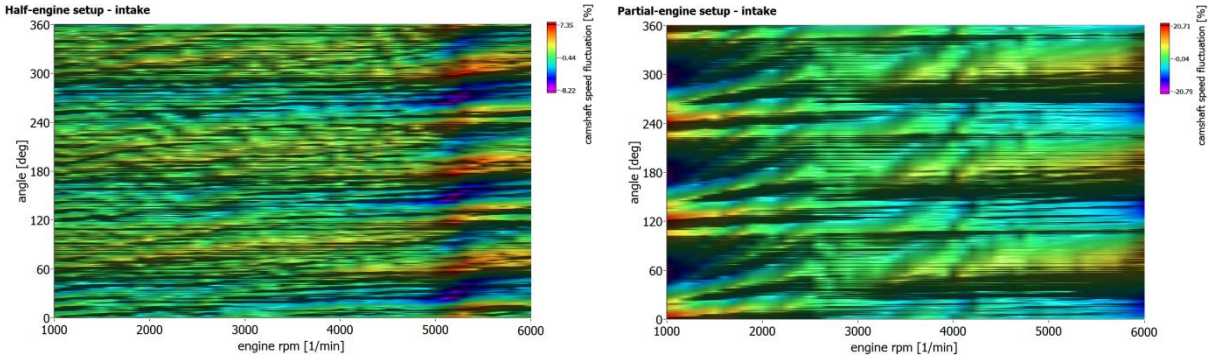


Figure 11.10 – Speed fluctuations, *standard* valvetrain, *Intake* camshaft, comparison of half (left) and partial (right) engine setup

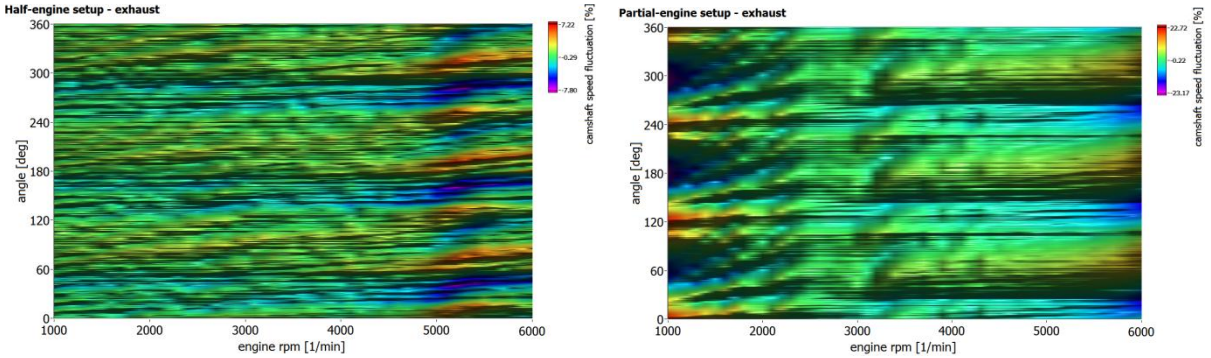


Figure 11.11 – Speed fluctuations, *standard* valvetrain, *Exhaust* camshaft, comparison of half (left) and partial (right) engine setup

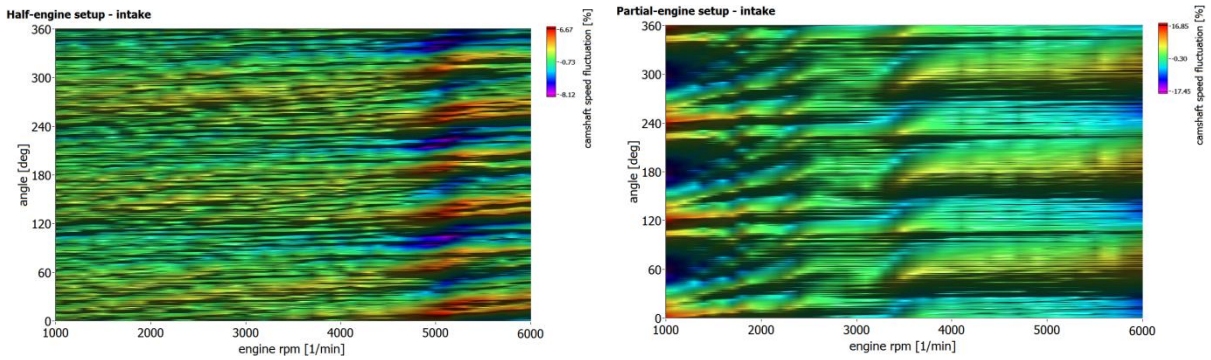


Figure 11.12 – Speed fluctuations, *low-friction* valvetrain, *Intake* camshaft, comparison of half (left) and partial (right) engine setup

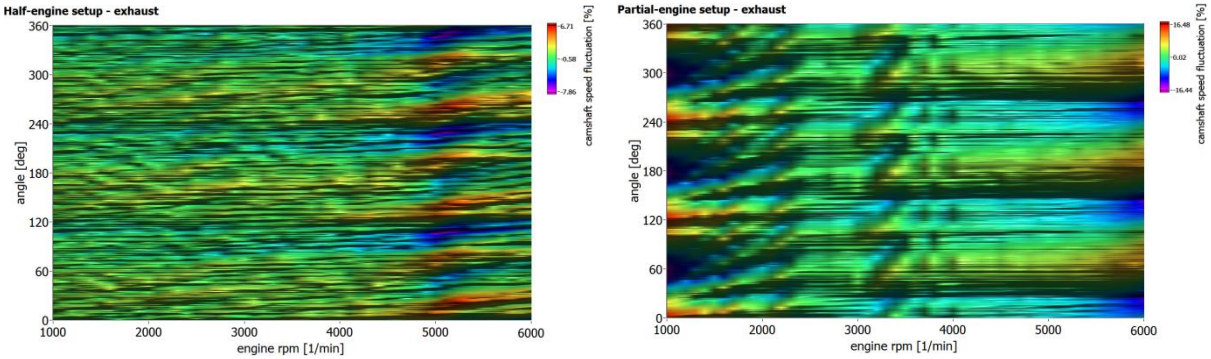


Figure 11.13 – Speed fluctuations, *low-friction* valvetrain, *Exhaust* camshaft, comparison of half (left) and partial (right) engine setup

11.3 Comparison of the kinematic variables

The most important comparison and reason why we carried out these measurements is the comparison of the kinematic variables. The question posed was whether the partial-engine setup delivered the same values of kinematic variables as the half-engine setup, i.e. whether the change in the speed fluctuations and missing parts of the valvetrain significantly change the valve behavior.

Figure 11.14 depicts the comparison of the valve kinematic variables of the first intake valve (closest to the drive, In_1) of the standard valvetrain. Figure 11.15 depicts the same for the last intake valve (In_6). The exhaust valves kinematics comparison is presented for the low-friction valvetrain (Figures 11.16 and 11.17) since the trends in behavior are the same for the standard and the low-friction valvetrain.

The acceleration was calculated using the central-difference formula of the 4th order that helps suppress the influence of the noise in this case:

$$f' = \frac{-f(x+2h)+8(x+h)-8(x-h)+f(x-2h)}{12h} \tag{11.1}$$

No difference of the valve displacement curves was noticed. Only small random offset drifts caused by resetting the laser probes and different levels of noise can be traced in the valve displacement data. There are noticeable differences in the maxima of the valve velocity,

which are caused by differences in the speed of rotation of each system and are explained in the previous chapter.

It can be concluded that in most cases, the measured variables of the partial-engine setup are fairly close to the variables of the half setup. However, there are some cases where the legitimacy of usage of the partial-engine system could be questioned, especially from the point of the valve acceleration. An example could be the data of In_1 at 6000 rpm (Fig 11.14). In this case, the half setup has a significantly higher amplitude of acceleration oscillations than what would be measured by the partial-engine setup. These data could lead to false conclusions about the necessary stiffness of the cam springs or behavior of the valvetrain. These differences are noticeable from approximately 5700 rpm and are most likely a result of the difference in the speed fluctuations between the two setups that continue increasing from 5500 rpm.

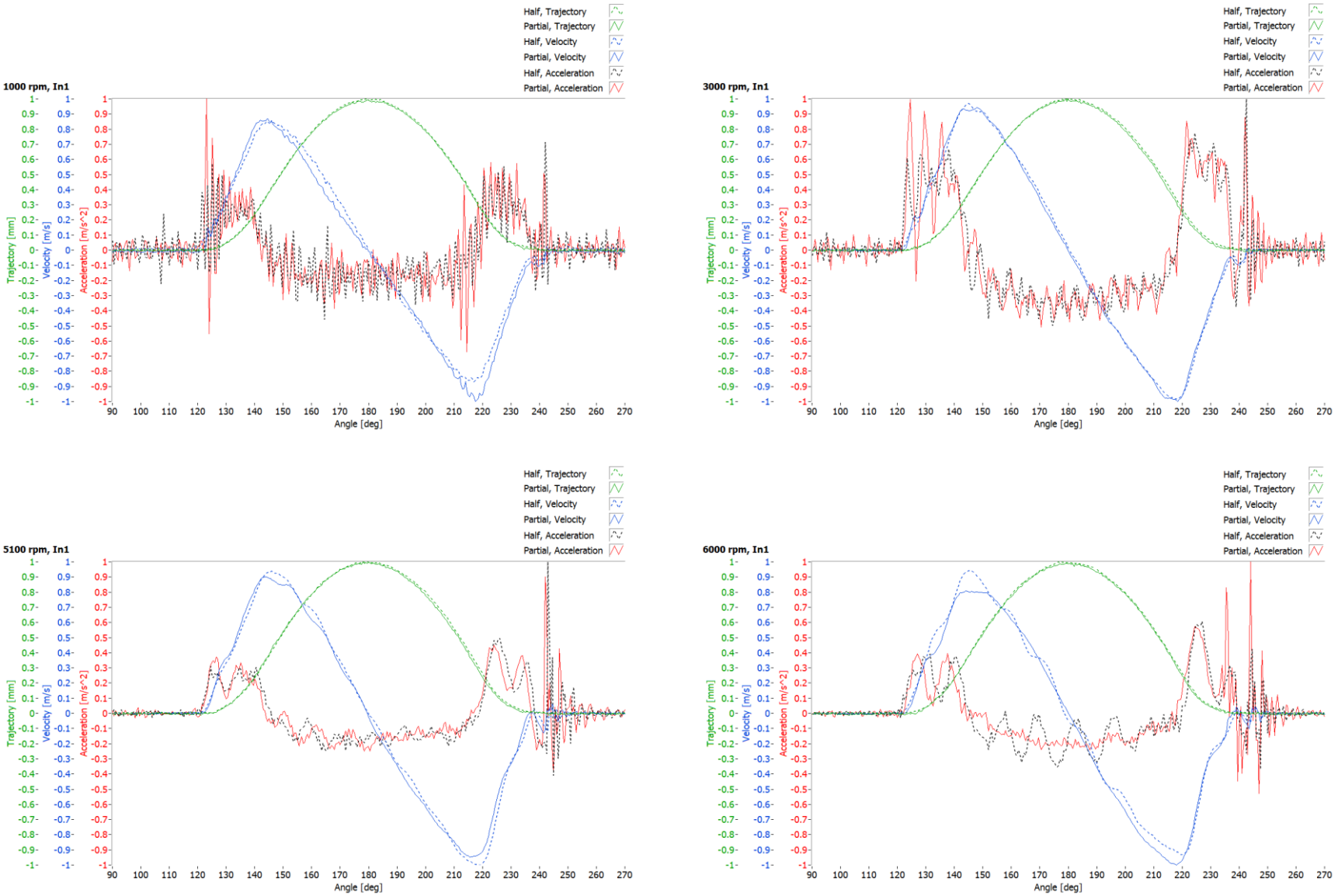


Figure 11.14 – *Standard* valvetrain, In_1, Comparison of valve kinematics of half-engine setup (dashed lines) and partial-engine setup (solid lines)

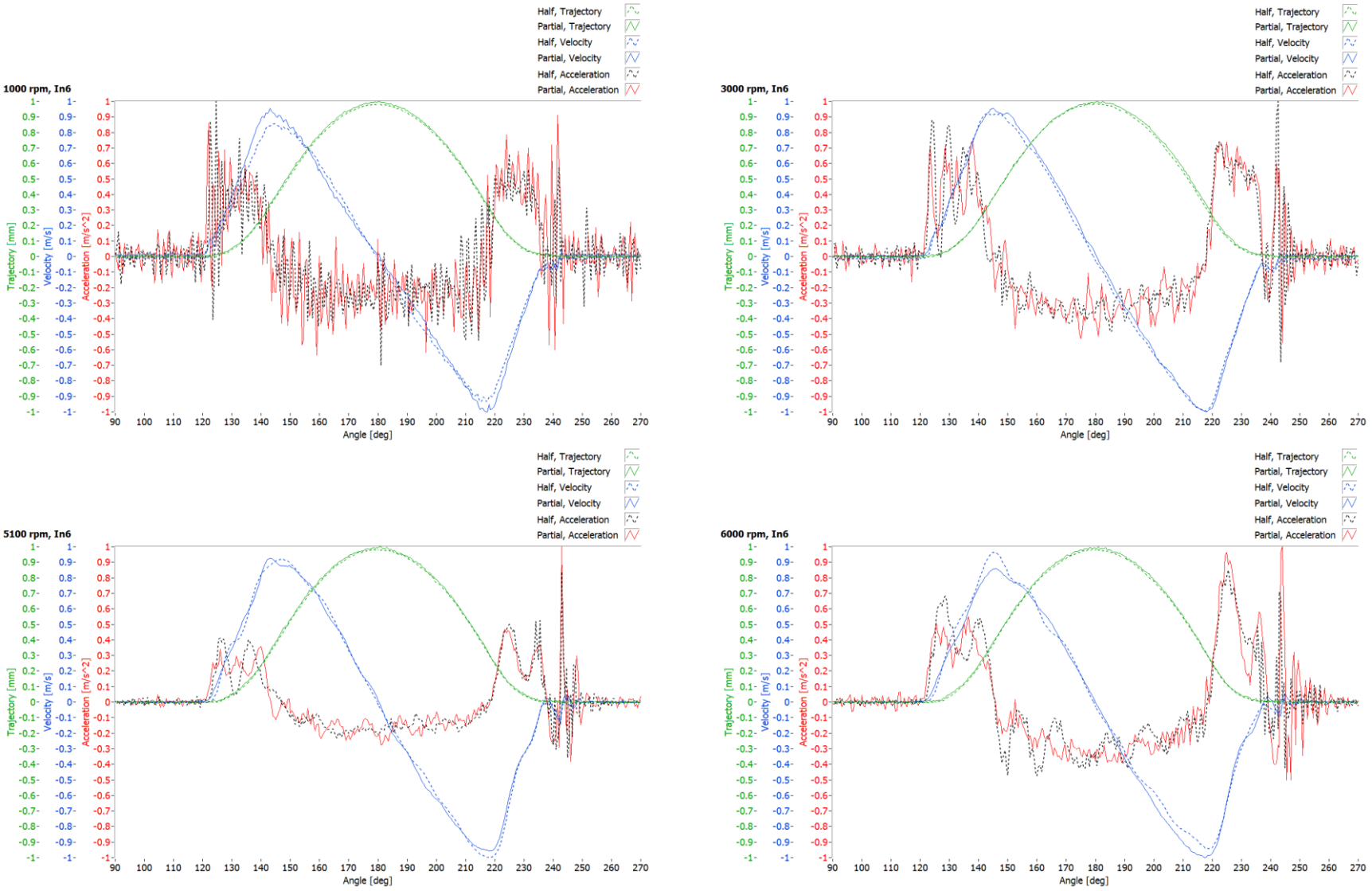


Figure 11.15 – *Standard* valvetrain, In_6, Comparison of valve kinematics of half-engine setup (dashed lines) and partial-engine setup (solid lines)

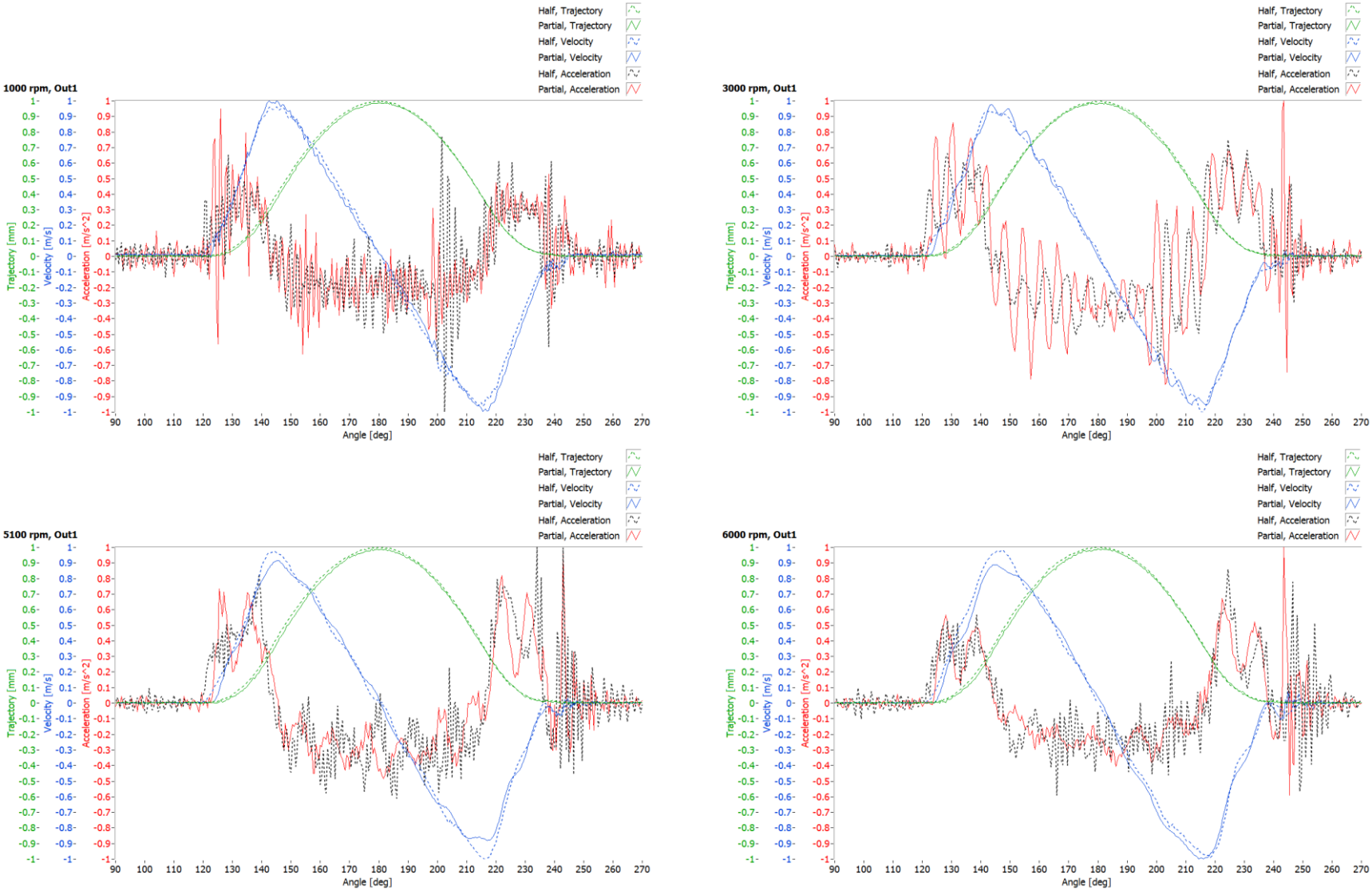


Figure 11.16 – *Low-friction* valvetrain, Out_1, Comparison of valve kinematics of half-engine setup (dashed lines) and partial-engine setup (solid lines)

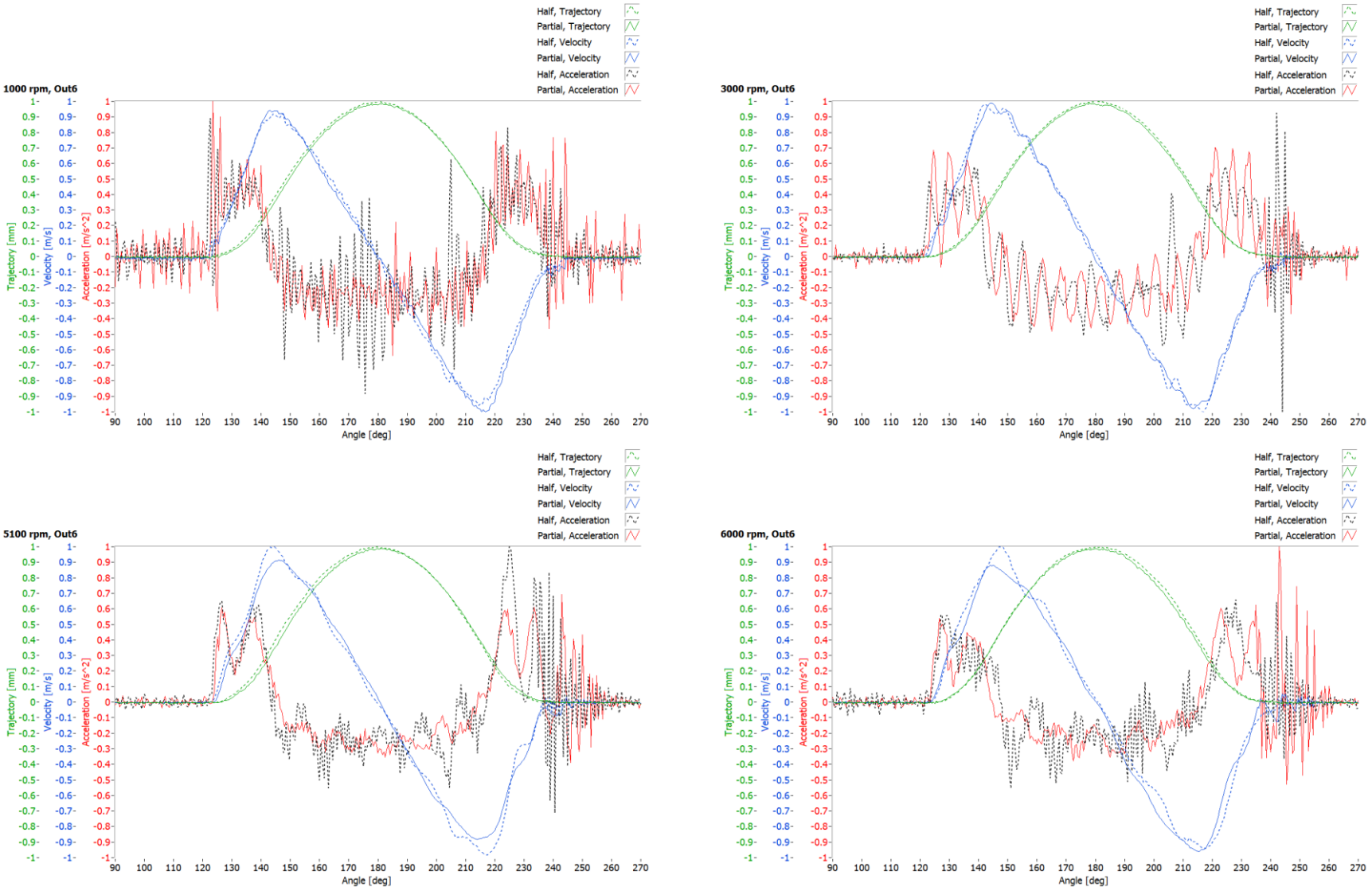


Figure 11.17 – *Low-friction* valvetrain, Out_6, Comparison of valve kinematics of half-engine setup (dashed lines) and partial-engine setup (solid lines)

11.4 Impact velocities

Last but not least, the parameter which offers a valuable comparison of the two setups is impact velocity – a crucial parameter which influences the lifetime of the engine. If the valve is repeatedly seated with a high velocity, the seat can be worn off and contact fatigue or other damage may appear. Due to resulting leakage, valve heat dissipation is not ensured and local heat load increases. The valve can even melt.

The following tables (Tables 11.3 and 11.4) and figures (Figures 11.18 and 11.19) present a comparison of the identified impact velocities. The impact velocity was evaluated from 4000 rpm to 6000 rpm, since lower speeds are not interesting for this analysis. Between 1000 rpm and 2000 rpm the impact velocities are very low and the exact point of impact is usually unidentifiable. All five recorded consecutive periods of data from each speed were examined in order to reduce the risk of overlooking a faulty valvetrain.

Since we are not permitted to publish the absolute values of the impact velocities the presented values were rescaled and should be interpreted as the percentage of the maximal impact velocity encountered in this thesis.

The valvetrain is considered functional if the impact velocity is less than 1 m/s (1000 mm/s) – the empirical parameter. All the impact velocities fulfill this criterion. No clear trend was identified in the differences between the impact velocities determined from the data measured by the partial-engine and the half-engine setup.

Table 11.3 – Impact velocities of standard valvetrain

	Impact velocity [% $v_{max\ impact}$]	*4000 rpm*	*5000 rpm*	*5100 rpm*	*5200 rpm*	*5300 rpm*	*5400 rpm*	*5500 rpm*	*5600 rpm*	*5700 rpm*	*5800 rpm*	*5900 rpm*	*6000 rpm*
In_1	Half-engine setup	39 36 38 38 39	75 72 70 57 74	68 74 75 78 70	64 59 58 53 59	72 68 74 71 72	45 72 64 63 62	32 62 68 62 66	64 76 71 72 72	54 32 46 43 36	46 39 43 51 39	51 61 51 36 55	42 46 51 50 47
	Partial-engine setup	34 38 36 34 36	68 58 74 53 61	62 74 74 45 72	74 76 74 76 79	45 86 78 39 58	62 47 74 82 89	82 74 66 66 74	70 71 70 70 47	66 64 83 80 59	- - - 49 70	74 - - 74 75	70 70 84 100 -
In_6	Half-engine setup	36 39 33 29 39	59 55 46 57 61	39 34 50 49 62	61 83 53 51 68	67 62 59 61 61	62 57 61 67 61	63 59 57 62 32	43 42 36 61 42	53 36 39 38 46	42 45 46 39 49	21 45 38 47 36	43 58 47 46 54
	Partial-engine setup	37 - 38 38 39	57 70 37 36 67	82 59 67 39 59	74 55 47 51 75	80 78 45 33 78	71 58 47 49 49	78 70 49 75 74	- 53 61 - -	42 63 49 39 54	32 29 72 72 22	57 59 37 51 43	62 70 57 74 63
Out_1	Half-engine setup	- - - 28 -	62 70 66 59 67	43 45 30 36 61	68 71 37 57 53	74 16 71 67 32	76 79 38 38 49	83 32 78 33 61	49 59 41 45 45	59 36 54 58 63	- 59 30 34 29	47 54 46 47 21	30 30 29 37 43
	Partial-engine setup	29 33 41 42 32	37 30 32 30 25	29 28 53 22 49	61 61 28 25 54	- - 20 46 33	49 53 - 17 30	57 82 - - 74	46 - - - - *(v<71)*	55 25 26 30 61	66 - - - 57	- 70 76 54 54	66 61 89 49 59
Out_6	Half-engine setup	30 28 32 28 29	- 54 - - -	41 - - - -	- - - - - *(v<39)*	38 - 53 - 53	- 61 50 34 55	34 - 41 32 43	- 51 58 42 -	- 43 - 42 37	32 32 37 28 53	32 28 - 30 28	- - - - - *(v<39)*
	Partial-engine setup	- 41 41 41 45	50 46 53 59 47	49 58 46 53 54	43 55 55 53 67	76 43 47 39 42	59 80 82 47 76	54 62 53 66 71	55 72 68 83 86	74 58 58 83 61	- 95 89 84 83	64 78 68 51 57	79 45 71 75 51

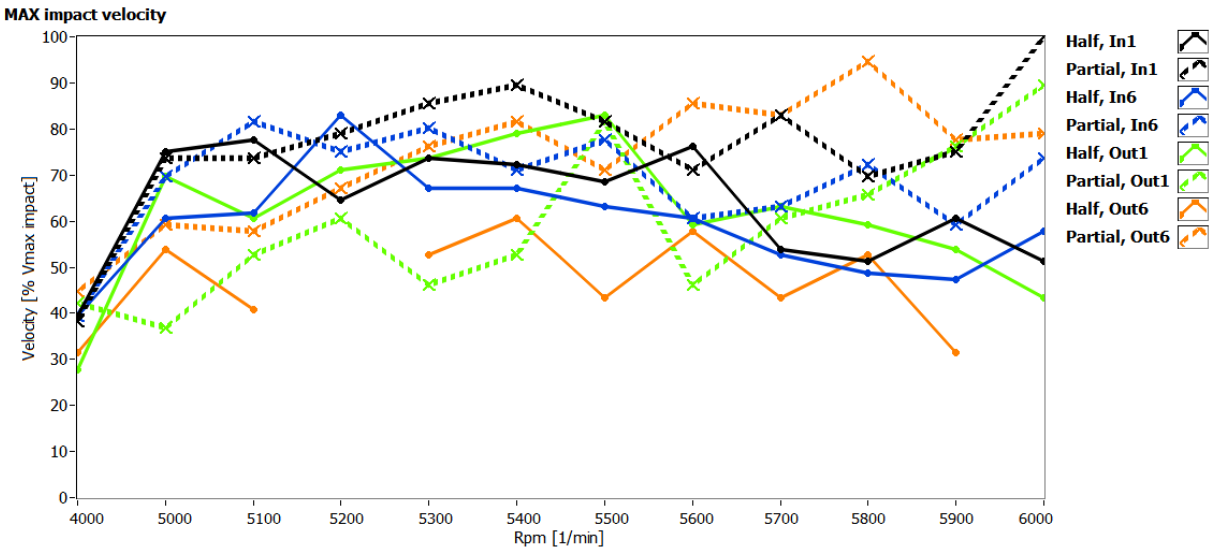


Figure 11.18 – Comparison of the maximal impact velocities of the *standard* valvetrain in half-engine setup (solid lines) and partial-engine setup (dashed lines)

Table 11.4 – Impact velocities of low-friction valvetrain

	Impact velocity [% $v_{max\ impact}$]	*4000 rpm*	*5000 rpm*	*5100 rpm*	*5200 rpm*	*5300 rpm*	*5400 rpm*	*5500 rpm*	*5600 rpm*	*5700 rpm*	*5800 rpm*	*5900 rpm*	*6000 rpm*
In_1	Half-engine setup	28 24 30 14 8	43 20 - 22 25	29 20 34 28 42	17 47 46 47 24	46 - 76 - -	- 57 47 28 55	37 54 45 51 -	- 42 - 32 -	14 - 13 36 20	- 32 - 45 45	21 12 34 18 34	13 11 16 24 26
	Partial-engine setup	17 18 20 24 20	53 39 34 38 39	43 37 - 38 -	47 29 49 33 47	45 64 68 70 59	47 67 67 51 42	41 49 - 72 66	39 72 55 - 28	42 76 71 47 47	34 71 58 45 75	36 79 43 57 -	46 76 58 - 68
In_6	Half-engine setup	45 34 39 36 34	37 53 37 34 51	28 41 30 33 34	- 24 50 43 -	46 - - 20 39	32 - 61 33 -	- 47 34 46 -	- 39 - - -	- - - 34 -	- - - - - (v<92)	- 59 - 63 -	Not measured
	Partial-engine setup	21 22 29 20 18	46 42 61 43 30	68 66 76 68 63	72 38 75 42 71	47 75 80 57 86	66 57 72 63 43	54 74 72 39 74	59 43 42 74 55	53 59 41 83 54	51 74 58 72 70	66 59 64 61 66	58 37 42 46 64
Out_1	Half-engine setup	39 34 - - -	- - - 43 50	- 39 47 - -	28 - - 30 -	38 41 41 61 46	- 46 - - -	- 38 32 - 54	22 26 47 38 33	39 36 50 25 54	63 54 37 42 49	24 33 41 37 47	55 - 33 22 39
	Partial-engine setup	3 - 28 17 3	34 22 42 20 33	21 46 37 38 -	36 46 45 - 22	41 24 34 14 33	38 38 53 57 46	37 59 - 63 38	- 42 51 42 38	- 37 58 25 21 38	59 38 28 39 34	59 38 28 39 34	64 45 74 - 67
Out_6	Half-engine setup	26 17 26 18 26	18 - 18 - -	39 - 42 - -	- - - - - (v<68)	- - 34 - - (v<100)	- - - 36 32	18 21 - 22 -	- - - - - (v<65)	- - 51 54 42	- - 29 43 51	13 32 55 - 37	17 34 - 13 - (v<61)
	Partial-engine setup	30 - 42 - 45	- - 39 32 34	20 - 39 28 22	38 32 33 41 28	41 - 42 21 22	24 29 29 25 54	- 22 64 43 43	53 49 34 - 17	61 57 61 58 57	71 67 67 76 55	58 30 74 62 82	- 76 36 78 29

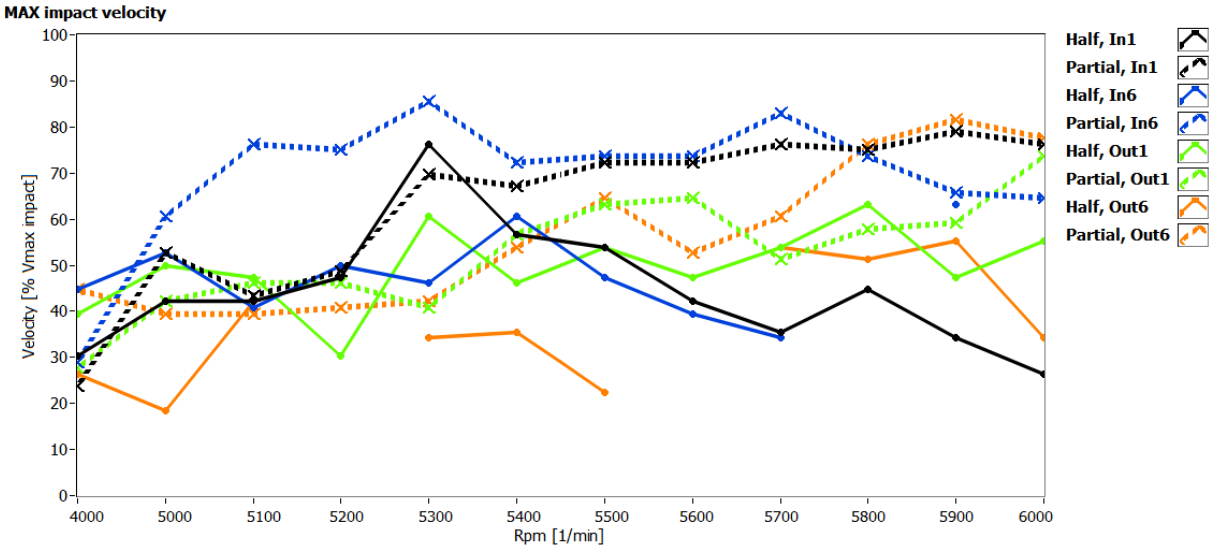


Figure 11.19 – Comparison of the maximal impact velocities of the *low-friction* valvetrain in half-engine setup (solid lines) and partial-engine setup (dashed lines)

12 Summary and discussion

The current measurement procedure of valve kinematics is time-consuming. The measurements are carried out over the given operational spectrum of the engine rpm. Higher frequencies tend to be more interesting for further processing, and thus are measured in detail with smaller rpm increment. At the same time it is necessary to have high throughput of the testbed. The aim of this work was to design, construct, and test a new generation of the apparatus for valve kinematics measurements, and to overcome the limitations of the current systems (namely time consumption, high level of manual tasks which could be automated, and saving of data affected by drop-out phenomenon).

At the beginning of this project, a multipurpose automated apparatus was constructed and has been improved in the last four years. Like the previous systems, it takes advantage of the non-contact nature of high-speed Laser Doppler Vibrometry (LDV) which has become a standard measurement technique for obtaining the displacement, velocity and acceleration of the valves. The designed apparatus spans all of the engine's operational speeds and can operate a half-engine or a partial-engine setup (the latter which utilizes only the head and the head cover and the related valvetrain components). The control software (developed in LabVIEW) automatically compensates for slippage of the ribbed belt that is used to transfer the torque between the shaft of the electromotor and the combustion engine. Moreover, it is capable of a truly parallel measurement of camshaft speed fluctuations, which helps to verify the proper function of the valvetrain components. Most of all, with the algorithm for LDV drop-out noise detection, the system is fully automated. It runs the measurement across predefined engine speeds, repeating if drop-out noise is detected and saving only the representative data for further processing.

During the first development stage two communication protocols for driving the electromotor were implemented: Siemens USS protocol and Lenze LECOM protocol. Motivation for the creation of the USS protocol API was the nonexistence of any implementation of the protocol into LabVIEW. As both developed libraries were given for public use, other users had already reported their usability. The resulting automated application now benefits from the implementation of both protocols, supporting a wide palette of different controllers which drive the engine through the connected electromotor.

A detail study of the available frequency measurement methods which utilize the counters of programmable counter boards was carried out. *Method 1 – Inverse period measurement* was selected for measurements where IRC pulse per revolution (ppr) is set to 720. For higher frequencies, e.g. for ppr = 3600, we selected *Method 3 – Measure time of known number of cycles* with divider = 4. Method 1 (M1) offers the frequency measurement error of 0.03% and Method 3 (M3) of 0.04% (at rpm = 4500). The application dynamically switches between the two if needed. We carried out tests to measure speed fluctuations of revolving shafts and discovered the precision of the IRC sensor to be the limiting factor for the overall accuracy. The worst error of the one-speed sample reading (between two consecutive IRC pulses) given by the accuracy of the IRC is 4% with M1 and ppr = 720 and 5% with M3 and ppr = 3600. We also outlined the half-width size of the moving average filter for both the methods to improve the measured results: M1 = 3 samples per 720 samples; M3 = 4 samples per 3600 samples and divider 4.

The main problems which arise during automation of the LDV measurement are *speckle noise* and *signal drop-outs.* A speckle pattern is produced when the coherent waves of the incident laser beam are dephased during backscatter from a surface that is rough on the scale of optical wavelength. The scattered yet still coherent waves interfere constructively and destructively, producing a chaotic distribution of light and dark spots. The speckle pattern is

not of high significance unless it changes dynamically. It is then able to translate (i.e. speckles appear to move in space while retaining their size and shape) or boil (i.e. no translation of the speckle but a continuous evolution from one size and shape to another). Speckle noise is produced if the Doppler signal amplitude remains high enough for the demodulator to operate. If the Doppler signal amplitude drops to low levels and the demodulation process fails, signal drop-out occurs. During its motion the valve is designed to rotate in order to keep the valve face and seat clean of carbon deposits. This also has the effect of slightly reducing the wear. Due to acting forces the valve also experiences tilt – especially at high speeds. All of these factors contribute to significant speckle noise during the valve kinematics measurement, as well as frequent drop-out noise.

We created a set of routines which were aimed at the recognition of signal drop-outs in the valve kinematics data. The developed algorithm helps to increase throughput of the engine test stand and decrease the time needed for evaluation of the valvetrains. The routines were tested on data which were collected for over 20 years and on data measured on combustion engines in our laboratory. The tests show satisfactory results in the identification of the phenomenon. Small drop-outs which are recognizable by the human eye might be on the edge of the algorithm's capabilities, especially if their size is not far from the estimated level of noise. The next step is to use the tests with different engines during everyday measurements in the ŠKODA AUTO laboratory, and to adjust the parameters if needed.

Thanks to the drop-out recognition tests, the measurement can be fully automated and repeated if necessary. The original measurement time of one valve has been shortened by 75%. Additional time saving is created by avoiding the manual repositioning of the IRC sensor if the IRC reference pulse would be generated during the opening/closing phase of the valve. The reference mark is supposed to trigger the measurements and reset the laser probes controller at every revolution. If it was generated during the opening/closing phase of the valve, it would start the acquisition at an unwanted position; and resetting the probes at this point would cause false reading. Our solution utilizes one programmable counter of the counter board for pulse generation. When the original trigger (reference pulse) arrives, it generates a new pulse with a predefined delay. The delay is specified as the amount of ticks of the source signal which is the IRC TTL pulse train. The operation is retriggerable.

To verify the proper function of the constructed apparatus and to adjust the parameters of the drop-out recognition tests, as well as to analyze and compare measured data of two different valvetrains, we measured the standard ŠKODA 1.2 HTP valvetrain and the low-friction version of the same valvetrain. Additional motivation was to verify the benefit of having the true camshaft speed fluctuations record for the valve acceleration computation. Common practice is to neglect the variance in rpm_{cam} and presume the rotational speed to be constant. The main reason is that the synchronized information about camshaft speed fluctuations is not usually available.

The achieved results were satisfactory proving the overall capabilities of the apparatus and the drop-out recognition tests. It is a great benefit to have the synchronized camshaft speed fluctuation information without utilizing another device. It is capable of revealing speeds critical for the valvetrain, (e.g. 5100 rpm for the ŠKODA 1.2 HTP valvetrain in a half-engine setup as a result of resonance of the timing chain). The performed analyses show two perfectly-working valvetrains with no malfunctions. The low-friction valvetrain achieves the same results with lower mass, different material of the cam lobes, thinner valve springs, smaller stiffness and different bearings. It operates with lower friction losses and thus requires less input power. An engine with such a valvetrain will have lower fuel consumption and will produce less emissions.

No valve float or valve bouncing was identified in either of both the measured data. The impact velocities, which is a crucial parameter for a valvetrain lifetime, were in the expected

range. The benefit of having the speed fluctuations data for the valve acceleration computation was not proved. There are insignificant differences between the acceleration obtained by presuming the speed of the shaft to be constant and by utilizing the speed fluctuations data. It is safe to say that in the case of modern valvetrains, rotational irregularities do not play a significant role in the valve acceleration calculation and the rotation can be simplified as a constant for this purpose.

The last part of our work was to carry out a comparison of the data measured by a *half-engine setup* and a *partial-engine setup,* which is preferred in industry. Instead of testing the complete engine, assembly of merely the head and head cover is used. Such a setup is easy to assemble and does not demand additional modifications of the components. At the same time the exchange of the valvetrain components is fast and the valves are easily optically accessible. On the other hand, it poses only a fraction of the original mass of the engine. Nevertheless, it was expected to deliver results close to the half-engine setup and thus close to the values of a real operating engine. Although the partial setup has become a widely used solution since the OHC valvetrains became standard, no direct comparison of the same valvetrain in half-engine and partial-engine setup has been published (to the author's knowledge). We conducted this comparison on a 3-cylinder ŠKODA 1.2 HTP engine first with a standard valvetrain and then with the low-friction version. The main focus was on the valve kinematic variables, but the two setups were also compared from the point of camshaft speed fluctuations.

It can be concluded that in most of the cases, the measured valve kinematics data of the partial-engine setup correlate with the data measured on the half-engine setup. However, there are some cases where legitimacy of the usage of the partial system could be questioned. At high speeds (5700-6000 rpm), the amplitude of the acceleration oscillations is significantly lower than what would be measured with the half-engine setup. It could lead to false conclusions about the necessary stiffness of the valve springs and valve behavior. These differences are most likely a direct result of the differences in the amplitude and phase of the speed fluctuations between the two setups.

The determined impact velocities do not significantly differ between the partial and the half-engine setup. From this point the partial-engine setup delivers correlated results.

As expected, the speed fluctuations of the two setups are very different. The higher camshaft speed fluctuations of the partial-engine setup are a result of lacking moving parts (and thus their inertia) and by driving only one camshaft at a time. To identify the exact origin of the differences in the speed fluctuations (amplitude, phase, orders) between the two setups we would need precise numerical models. To construct them is beyond the scope of this thesis and leaves door open for future research.

These measurements should be a significant source of information for the engineers in automotive industry and should contribute to the ongoing discussion about the modifications of the partial-engine setup. The updated version will most likely add a timing chain/timing belt in order to drive both the sprockets of the camshafts. The chain tensioner blade and a ‘dummy’ crankshaft sprocket will be positioned exactly as in a real-engine. Instead of driving the camshaft the crankshaft sprocket/pulley will be driven by the electromotor. The consequent measured valve kinematics should be closer to the values of a real engine.

References

[1] Marcello Montanari, Fabio Ronchi, Carlo Rossi, and Alberto Tonielli, "Control of a Camless Engine Electromechanical Actuator: Position Reconstruction and Dynamic Performance Analysis," *IEEE transactions on industrial electronics*, vol. Vol 51, no. 2, April 2004.

[2] Y. Wang, A. Stefanopoulou, K. Peterson, T. Megli, and M. Haghgooie, "Modeling and control of electromechanical valve actuator," *SAE Paper 2002–01–1106*, 2002.

[3] D. Kim, M. Anderson, T. Tsao, and M. Levin, "A dynamic model of a springless electrohydraulic camless valvetrain systems," *SAE Paper 970 248*, 1997.

[4] Fiat Group Automobiles S.p.A., "Adaptive control system of the air-fuel ratio of an internal combustione engine with a variable valve timing system," EP 2 204 566 A1, Dec. 29, 2008.

[5] T. D. Choi, O. J. Eslinger, C. T. Kelley, J. W. David, and M Etheridge, "Optimization of Automotive Valve Train Components with Implicit Filtering," *Optimization and Engineering*, vol. Volume 1, Number 1 / June, 2000, pp. 9-27, June 2000.

[6] Andrea Carlini, Allessandro Rivola, Giorgio Dalpiaz, and Alberto Maggiore, "Valve motion measurements on motorbike cylinder heads using high-speed laser vibrometer," in *Proc. SPIE Vol. 4827, Fifth International Conference on Vibration Measurements by Laser Techniques:* , Ancona, Italy, Tuesday 18 June 2002, pp. p. 564-574.

[7] Petr Hošek, Analysis of kinematic variables of valve train, diploma thesis (in Czech), 2006, TUL.

[8] M. Gasparetti, N. Paone, and E. P. Tomasini, "Laser Doppler techniques for the combined measurement of inlet flow and valve motion in IC engines," *Measurement Science and Technology*, no. Volume 7, Number 4, April 1996.

[9] B. F. Payne and Charles Federman, "An automated fringe counting laser interferometer for low frequency vibration measurements," *International Instrumentation Symposium*, pp. p. 1-7, 1986.

[10] H. Selbach, A.C. Lewin, and V. Roth, "Laser Doppler-vibrometer for applications in the automotive industry," in *Proceedings 25th ISATA Silver Jubilee International Symposium on Automotive Industry*, Florence, Italy, 1992.

[11] S.J. Rothberg and Halliwell, N.A. Baker, "Laser vibrometry: pseudo-vibrations," *Journal of Sound and Vibrations*, no. 135(3), pp. 516-522, 1989.

[12] Peter Martin and Steve Rothberg, "Introducing speckle noise maps for Laser Vibrometry," *Optics and Lasers in Engineering*, no. 47, pp. 431-442, 2009.

[13] Peter Martin and Steven, J. Rothberg, "Pseudo-vibration sensitivities for commercial laser vibrometers," *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2753-2765, October 2011.

[14] Harold Rothbart, *Cam Design Handbook: Dynamics and Accuracy*, 1st ed.: McGraw-Hill Professional, October 23, 2003.

[15] Kishiro Akiba and Toshiaki Kakiuchi, "A Dynamic Study of Engine Valving Mechanisms: Determination of the Impulse Force Acting on the Valve," *A Dynamic Study of Engine Valving Mechanisms: Determination of the Impulse Force Acting on the Valve*, vol. vol. 3, pp. 6659-6664, 1988.

[16] ŠkodaAuto a.s., Učební pomůcka motor 1,4 l / 44kW, 2001.

[17] ŠkodaAuto a.s., Učební pomůcka motory 1,4 l - 16 V 55 kW a 1,4 l - 16 V 74 kW, 2005.

[18] ŠkodaAuto a.s., Učební pomůcka motor 2,0 l / 85kW a 88kW, 2005.

[19] ŠkodaAuto a.s., Učební pomůcka přeplňovaný motor 1,8 l / 110 kW - 5 ventilů, 2003.

[20] A. P. Pisano and F. Freudenstein, "An Experimental and Analytical Investigation of the Dynamic Response of a High-Speed Cam-Follower System. Part 1: Experimental Investigation," *Journal of Mechanisms, Transmissions and Automation in Design*, no. 105, pp. pp.692–698, 1983.

[21] Irving M. Gottlieb, *Electric Motors and Control Techniques*, 2nd ed.: McGraw-Hill Professional Publishing, February 1994.

[22] National Instruments, NI 6115/6120 User Manual, November 2002.

[23] National Instruments, DAQ-STC Technical Reference Manual, 1995, Part Number 340934A-01.

[24] Polytec, OFV-3001 Laser Vibrometer Controller (Datasheet).

[25] L. E. Drain, *The Laser Doppler Technique*.: John Wiley & Sons, October 1980.

[26] F. Durst, A. Melling, and J.H. Whitelaw, *Principles and practice of laser-Doppler anemometry*, 2nd ed., 1981.

[27] Siemens AG, Specification, USS protocol, 1994.

[28] Siemens AG, CU240S Operating Instructions, 10/2007.

[29] Siemens AG, SINAMICS G120, Control Units CU240S, Parameter manual, 2005.

[30] Natioanl Instruments, 6601/6602 User Manual, January 1999.

[31] National Instruments. (2007, Oct 5) NI Developer zone - Making Accurate Frequency Measurements. [Online]. http://zone.ni.com/devzone/cda/tut/p/id/3619

[32] National Instruments. (2009, Apr.) NI Developer zone - At What Signal Rates Can the Data Acquisition DAQ-STC Counter Chip Acquire? [Online]. http://digital.ni.com/public.nsf/allkb/DB428A46F345DA4986256AA100556876

[33] K. W. Wang, "On the Stability of Chain Drive Systems Under Periodic Sprocket Oscillations," *Journal of Vibration and Acoustics*, p. 119 (8 pages), January 1994.

[34] Douglass Bruce Powel, "Class 505/525: State machines and Statecharts," in *Proceedings of Embedded Systems Conference*, San Francisco, Fall 2001.

[35] Henk C. Tijms, *Algorithmic Analysis of Queues, Chapter 9 in A First Course in Stochastic Models*. Chichester: John Wiley & Sons, Ltd., 2003.

[36] National Instruments, NI Developer Zone - Queued State Machines, Feb 1, 2006.

[37] Anthony Lukindo. (2007, October 1) LabVIEW queued state machine architecture. LabVIEW and Visual Object-Oriented Programming Blog.

[38] NI Discussion Forums. (04-08-2007 08:37 AM) NI Discussion Forums. [Online]. http://forums.ni.com/t5/LabVIEW/Community-Nugget-4-08-2007-Action-Engines/td-p/503801

[39] National Instruments. (2008, Nov 28) Differences Between Multithreading and Multitasking for Programmers. Multicore Programming Fundamentals Whitepaper Series.

[40] Jeffrey Travis and Jim Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*, 3rd ed.: Prentice Hall, August 6, 2006.

[41] N. Takai, T. Iwai, and T. Asakura, "Correlation distance of dynamic speckles," *Applied Optics*, vol. 22, no. 1, pp. 170-177, 1983.

[42] Takashi Nakamura and Toshimitsu Asakuraa, "Statistical properties of dynamic speckles produced by a curved surface," *Optics Communications*, vol. 98, no. 4-6, pp. 331-339,

May 1993.

[43] Victor Hillier and Peter Coombes, *Fundamentals of motor vehicle technology*, 5th ed.: Nelson Thornes.

[44] G. Agostinelli, C. Cristalli, N. Paone, and S. Serafini, "Drop-out noise of laser vibrometers measuring on varnished steel surfaces of appliance cabinets for industrial diagnostics," in *9th International conference on vibration measurements by laser and non-contact techniques and short course*, May 28, 2010, pp. 298-312.

[45] J. Vass, R. Šmíd, R.B. Randall, P. Sovka, C Cristalli, and B Torcianti, "Avoidance of speckle noise in laser vibrometry by the use of kurtosis ratio: Application to mechanical fault diagnostics," *Mechanical Systems and Signal Processing*, vol. 22, pp. 647–671, 2008.

[46] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, no. 11, pp. 1-21, 1969.

[47] Shu-Ho Dai and Ming-O Wang, *Reliability analysis in engineering applications*. New York: Van Nostrand Reinhold, 1992.

[48] A. Savitzky and M.J.E Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[49] C. R. Meyer and H. N. Keiser, "Electrocardiogram baseline noise estimation and removal using cubic splines and state-space computation techniques," *Computers and Biomedical Research*, vol. 10, no. 5, pp. 459-470, October 1977.

[50] J. H. McClellan and T. W. Parks, "A Personal History of the Parks–McClellan Algorithm," *Signal Processing Magazine, IEEE*, vol. 22, no. 2, March 2005.

[51] G. Taylor and T. Campbell, "Design analysis of cam and tapped interaction for wear reduction in marine diesel engines," *Society of Automotive Engineering (SAE)*, vol. vol. 3, p. 8, 1989.

[52] M. Roskilly and et al., "Valve Gear Design Analysis," *Society of Automotive Engineering*, vol. vol. 3, 1986.

[53] P. Kreuter and G. Mass, "Influence of Hydraulic Valve Lash Adjusters on the Dynamic Behavior of Valve Trains," *Society of Automotive Engineering*, 1987.

[54] M. Husselman, Modeling and verification of valve train dynamics in engines, 2005, Phd Thesis.

Appendix 1

Frequency measurement error calculations
Matlab code

Method 1:

```
clear all
close all
clc
format bank
% --------------------

ppr=720; %3600
CrankRpm=5500;

timebase=80e6;
CamRpm=CrankRpm/2;
FCam=CamRpm/60;      %frekvence otaceni vacky [Hz]
TCam=1/FCam;         %perioda jedne otacky vacky [s]
FPuls=FCam*ppr       %frekvence pulzu pri danych otackach


RecPulseCount=timebase/FPuls      %kolik pulzu timebase je detekovano mezi
     dvema IRC pulzy
RecPulseCount=floor(RecPulseCount);
% --------------------

CalcFPuls0=(timebase/RecPulseCount)          %vypoctena frekvence pulzu pri
     danych otackach
CalcFCam0=CalcFPuls0/ppr;                      %vypoctena frekvence otaceni
     vacky
CalcCrankRpm0=CalcFCam0*60*2;                %vypoctene otacky kliky

CalcFPuls1=(timebase/(RecPulseCount+1))      %vypoctena frekvence pulzu pri
     danych otackach
CalcFCam1=CalcFPuls1/ppr;                      %vypoctena frekvence otaceni
     vacky
CalcCrankRpm1=CalcFCam1*60*2;                %vypoctene otacky kliky

CalcFPuls_1=(timebase/(RecPulseCount-1))     %vypoctena frekvence pulzu pri
     danych otackach
CalcFCam_1=CalcFPuls_1/ppr;                    %vypoctena frekvence otaceni
     vacky
CalcCrankRpm_1=CalcFCam_1*60*2;              %vypoctene otacky kliky


fx=FPuls;
fk=timebase;
Err=fx*fx/(fk-fx);
ErrRpm=(Err/ppr)*60*2   %klika

ErrProc=ErrRpm/(CrankRpm/100)
% --------------------
```

Method 2:

```
clear all
close all
clc
% --------------------
```

```
ppr=720; %3600
CrankRpm=5500;

Cbase=1;   %Signal from the first counter
CamRpm=CrankRpm/2;
FCam=CamRpm/60;      %frekvence otaceni vacky [Hz]
TCam=1/FCam;         %perioda jedne otacky vacky [s]
FPuls=FCam*ppr;      %frekvence pulzu pri danych otackach


RecPulseCount=floor(FPuls/Cbase)       %kolik pulzu timebase je detekovano
    mezi dvema IRC pulzy
% --------------------

CalcFPuls0=(Cbase*RecPulseCount);            %vypoctena frekvence pulzu pri
    danych otackach
CalcFCam0=CalcFPuls0/ppr;                       %vypoctena frekvence otaceni
    vacky
CalcCrankRpm0=CalcFCam0*60*2                %vypoctene otacky kliky

CalcFPuls1=(Cbase*(RecPulseCount+1))        %vypoctena frekvence pulzu pri
    danych otackach
CalcFCam1=CalcFPuls1/ppr;                       %vypoctena frekvence otaceni
    vacky
CalcCrankRpm1=CalcFCam1*60*2    %vypoctene otacky kliky

CalcFPuls_1=(Cbase*(RecPulseCount-1))       %vypoctena frekvence pulzu pri
    danych otackach
CalcFCam_1=CalcFPuls_1/ppr;                     %vypoctena frekvence otaceni
    vacky
CalcCrankRpm_1=CalcFCam_1*60*2              %vypoctene otacky kliky
% --------------------

err=FPuls/Cbase;
```

Method 3:

```
clear all
close all
clc
% --------------------

ppr=720; %3600
CrankRpm=5500;
divider=4;

timebase=80e6;
CamRpm=CrankRpm/2;
FCam=CamRpm/60;      %frekvence otaceni vacky [Hz]
TCam=1/FCam;         %perioda jedne otacky vacky [s]
FPuls=FCam*ppr       %frekvence pulzu pri danych otackach


RecPulseCount=timebase/(FPuls/divider)          %kolik pulzu timebase je
    detekovano mezi dvema IRC pulzy
RecPulseCount=floor(RecPulseCount)
% --------------------

CalcFPuls0=((timebase/RecPulseCount)*divider);                  %vypoctena
    frekvence pulzu pri danych otackach
CalcFCam0=CalcFPuls0/ppr;                       %vypoctena frekvence otaceni
    vacky
```

```
CalcCrankRpm0=CalcFCam0*60*2               %vypoctene otacky kliky

CalcFPuls1=((timebase/(RecPulseCount+1))*divider);               %vypoctena
     frekvence pulzu pri danych otackach
CalcFCam1=CalcFPuls1/ppr;                      %vypoctena frekvence otaceni
     vacky
CalcCrankRpm1=CalcFCam1*60*2               %vypoctene otacky kliky

fx=FPuls;
fk=timebase;
N=divider;
Err=fx*fx/(N*fk-fx)
ErrRpm=(Err/ppr)*60*2

ErrProc=ErrRpm/(CrankRpm/100)
% --------------------
```

Appendix 2

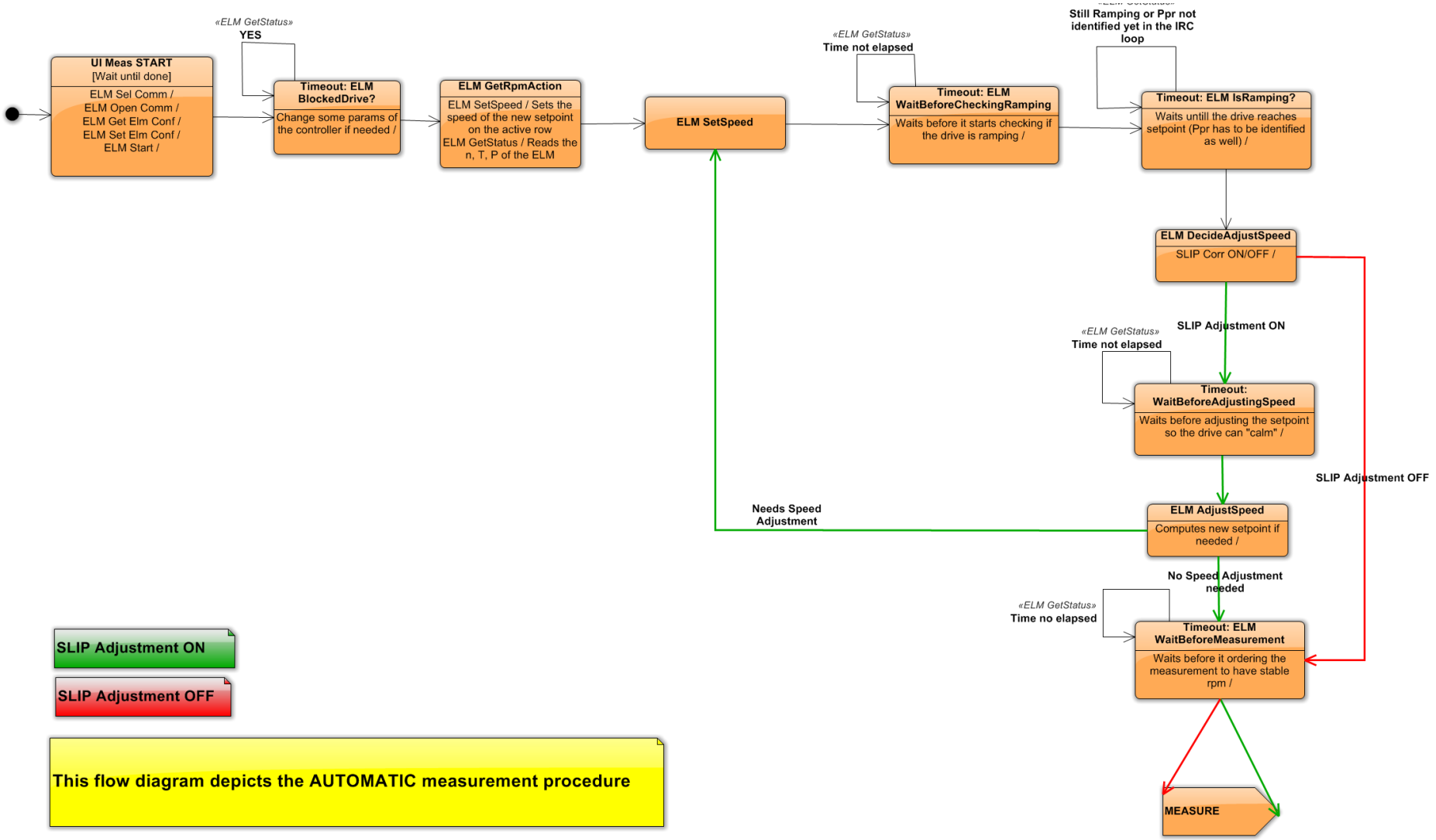
System parameters

Notebook with Intel® Core™2 Duo Processor@ 2GHz, 3GB RAM, Win 7-32bit, the cards in Magma rack + express card extension
DMA buffer 300.000 samples

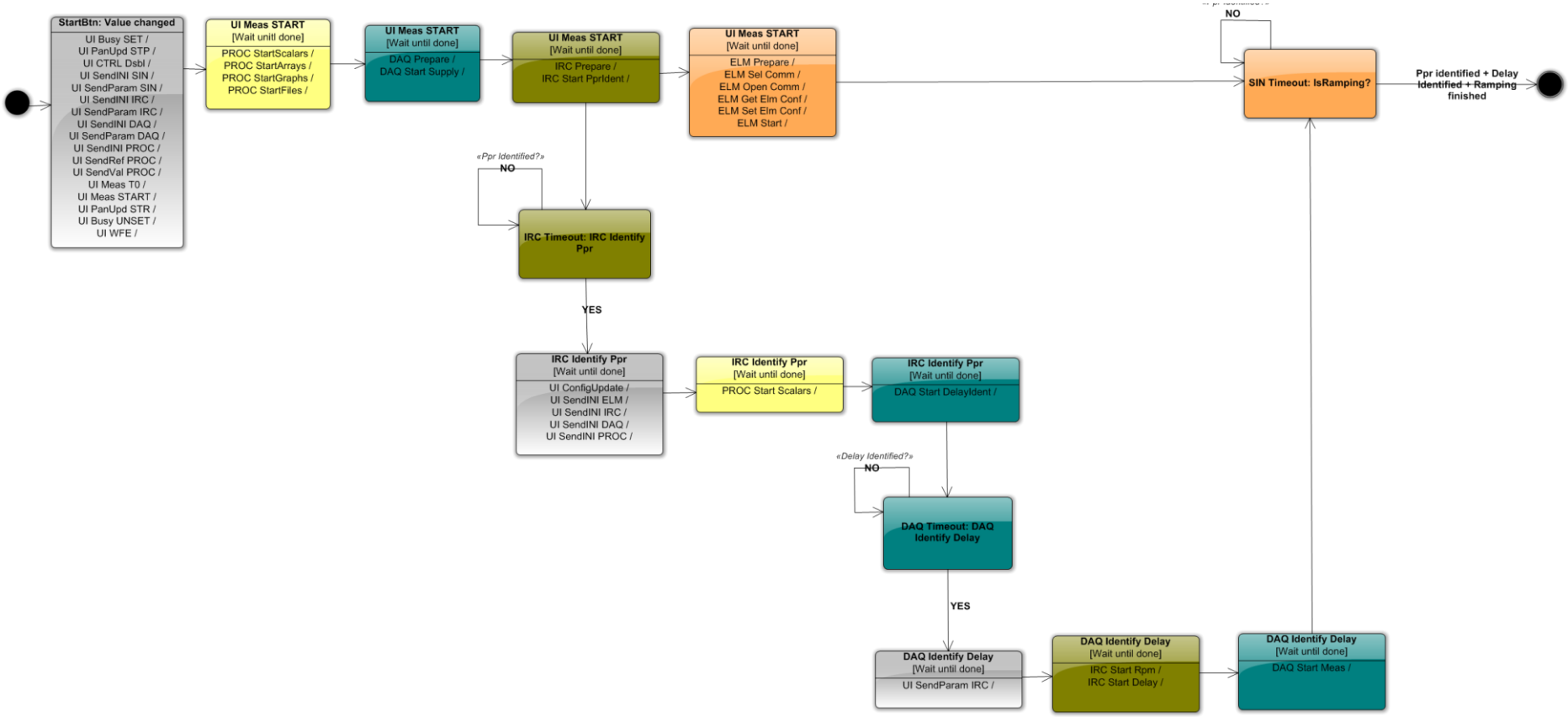
Appendix 3

DREAM application state charts and printscreens

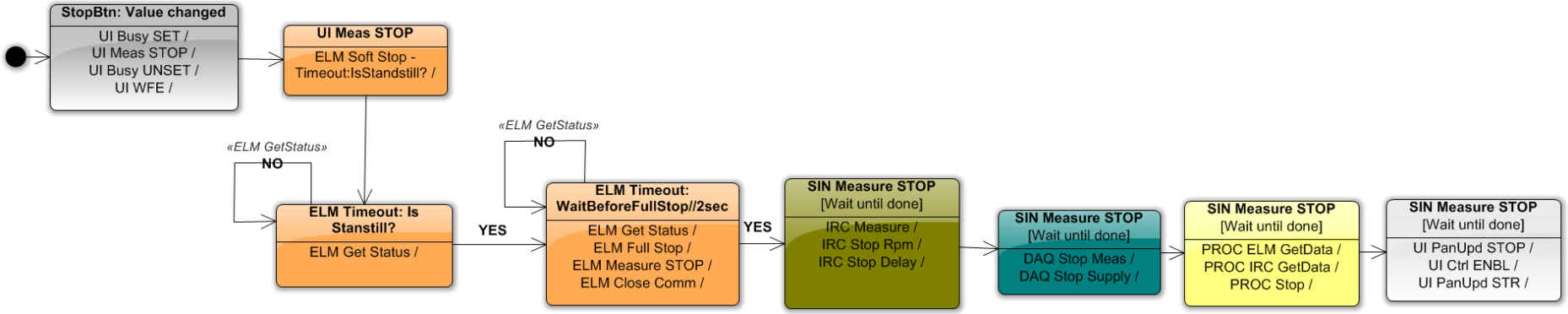
Run diagram:


UI Meas START
[Wait until done]
ELM Sel Comm /
ELM Open Comm /
ELM Get Elm Conf /
ELM Set Elm Conf /
ELM Start /
«ELM GetStatus»
YES
Timeout: ELM BlockedDrive?
Change some params of the controller if needed /
ELM GetRpmAction
ELM SetSpeed / Sets the speed of the new setpoint on the active row
ELM GetStatus / Reads the n, T, P of the ELM
ELM SetSpeed
«ELM GetStatus»
Time not elapsed
Timeout: ELM WaitBeforeCheckingRamping
Waits before it starts checking if the drive is ramping /
Still Ramping or Ppr not identified yet in the IRC loop
Timeout: ELM IsRamping?
Waits untill the drive reaches setpoint (Ppr has to be identified as well) /
ELM DecideAdjustSpeed
SLIP Corr ON/OFF /
SLIP Adjustment ON
«ELM GetStatus»
Time not elapsed
Timeout: WaitBeforeAdjustingSpeed
Waits before adjusting the setpoint so the drive can "calm" /
SLIP Adjustment OFF
Needs Speed Adjustment
ELM AdjustSpeed
Computes new setpoint if needed /
No Speed Adjustment needed
«ELM GetStatus»
Time no elapsed
Timeout: ELM WaitBeforeMeasurement
Waits before it ordering the measurement to have stable rpm /
MEASURE
SLIP Adjustment ON
SLIP Adjustment OFF
This flow diagram depicts the AUTOMATIC measurement procedure

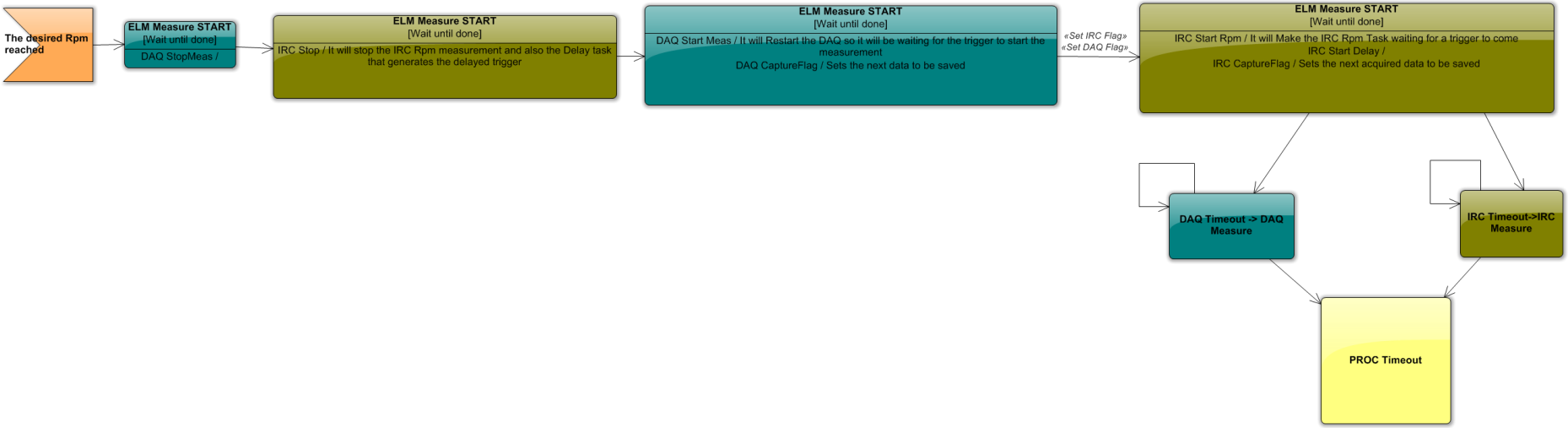
Start diagram:



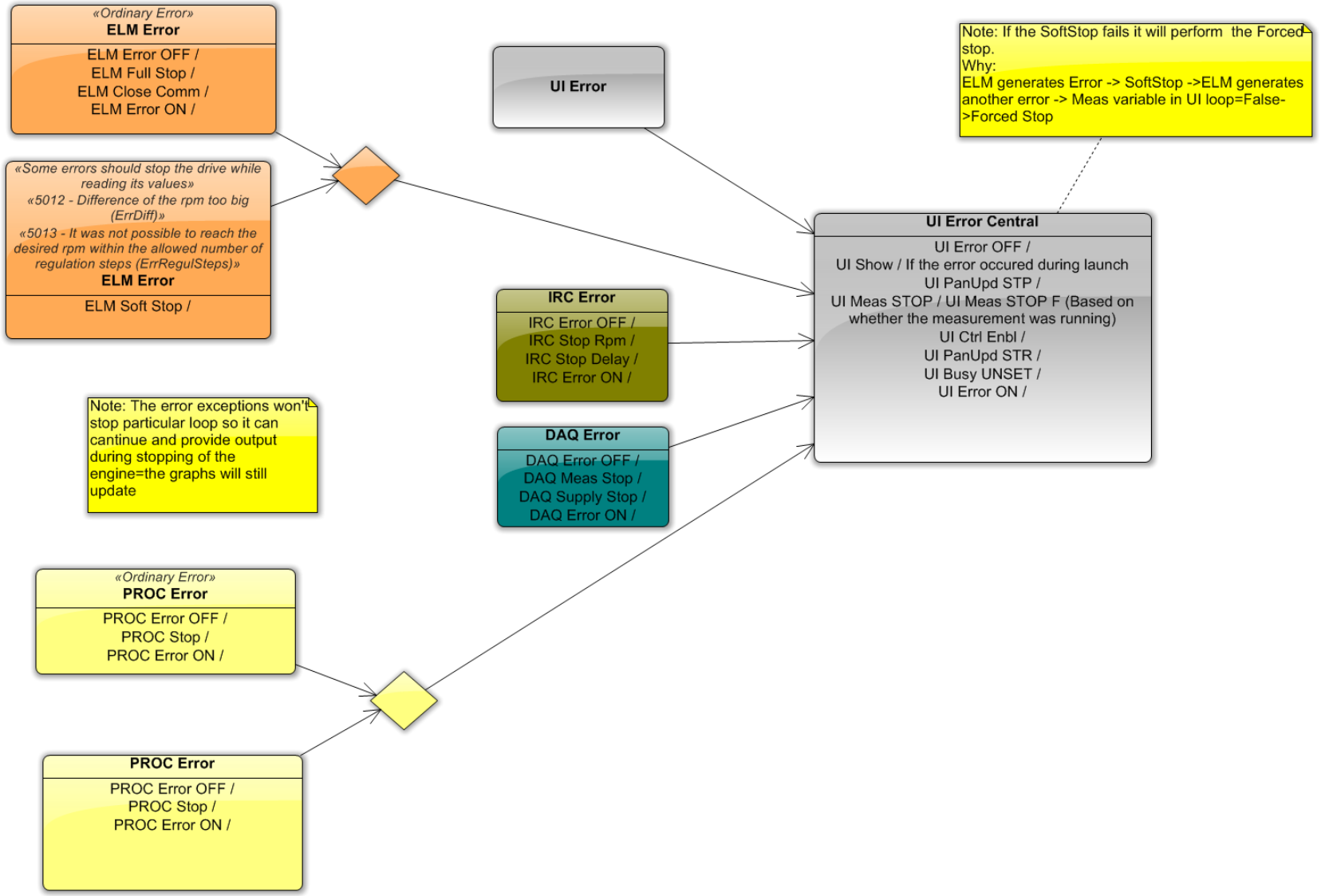
Stop diagram:



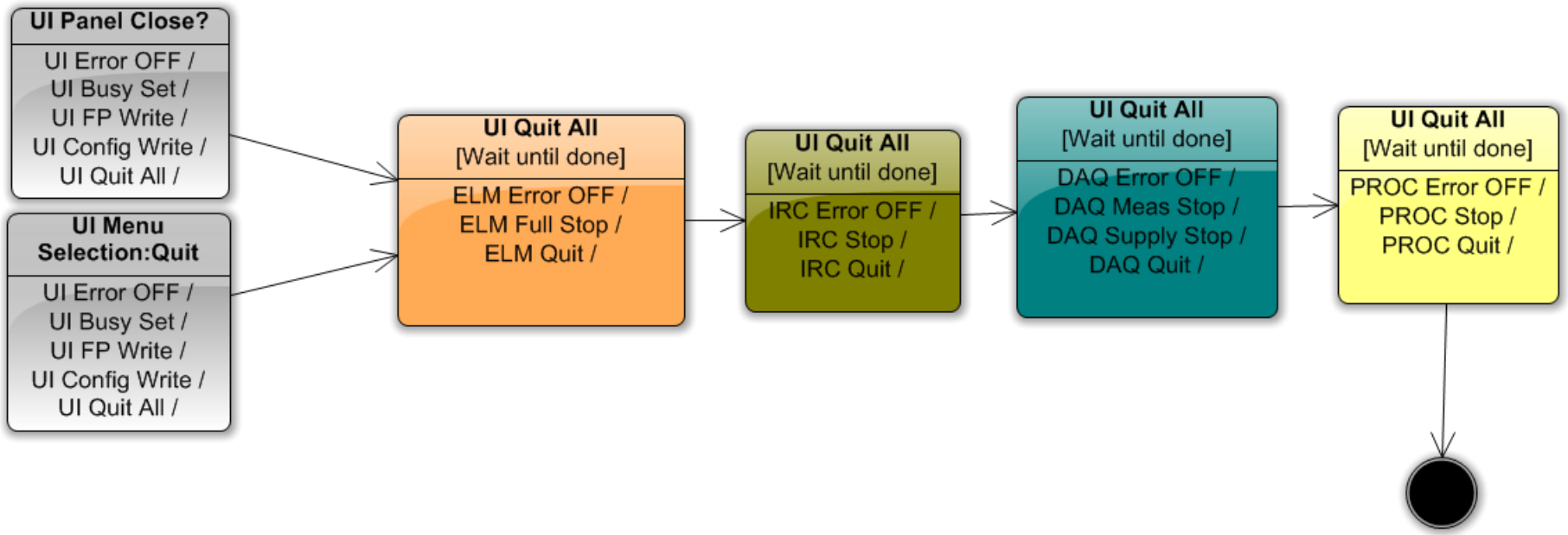
Measure diagram:



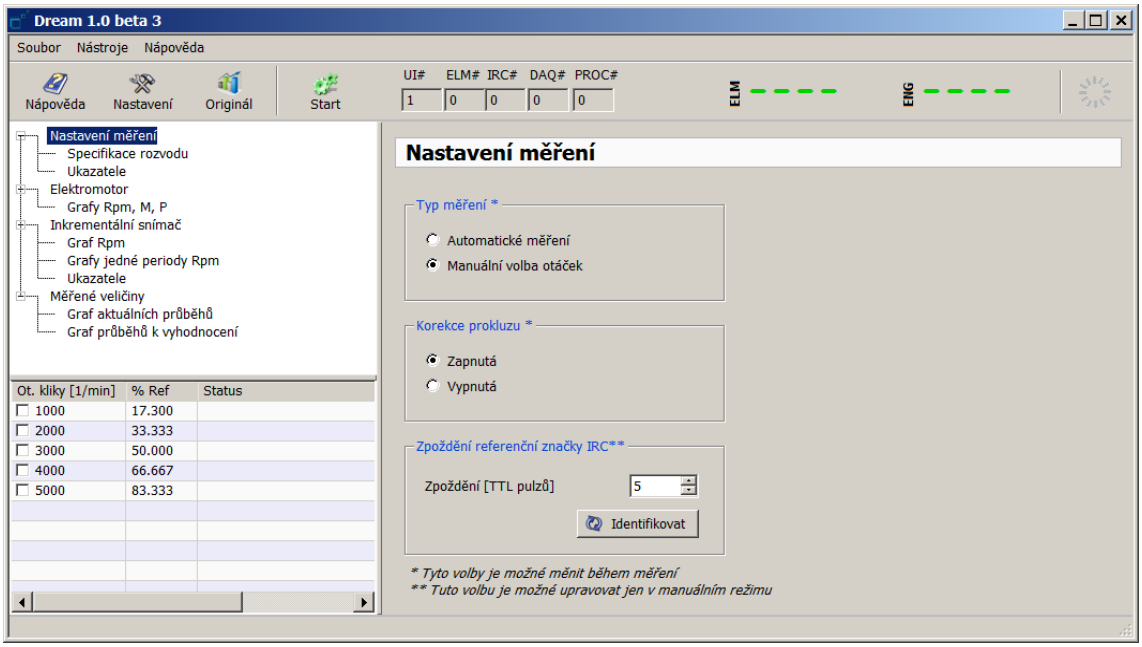
Error diagram:

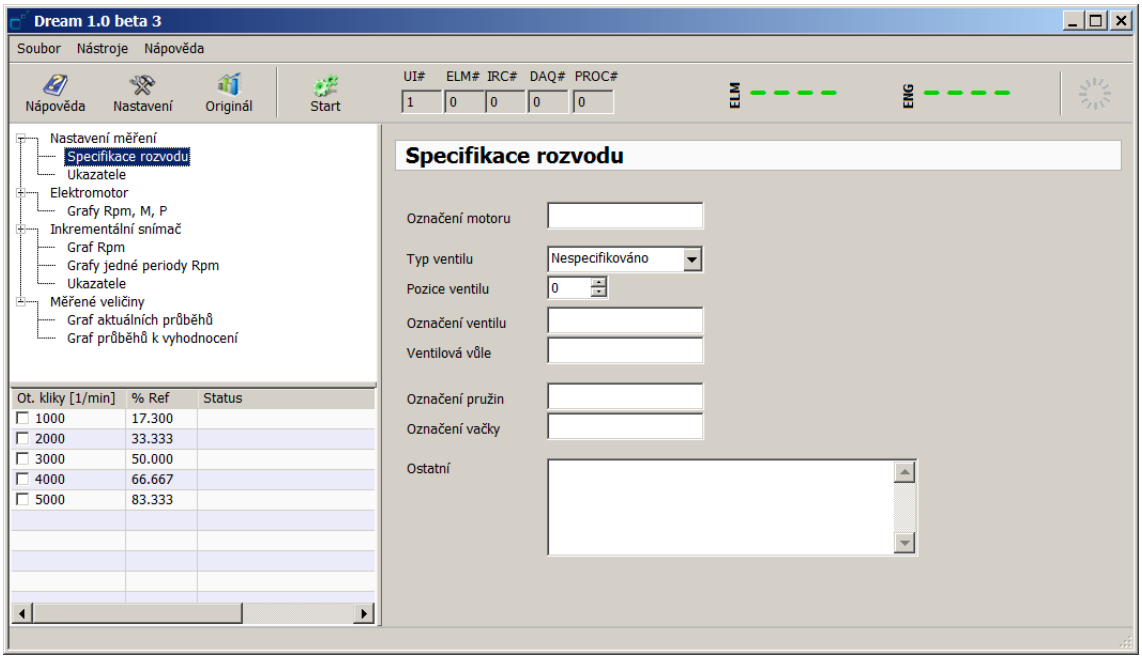


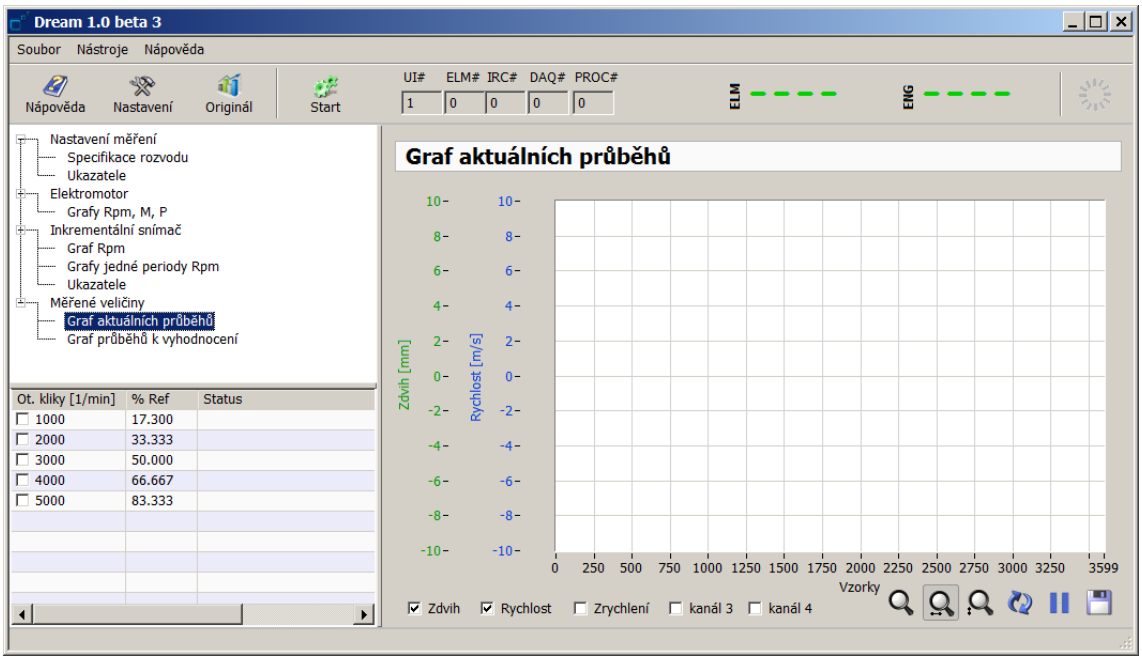
Quit diagram:

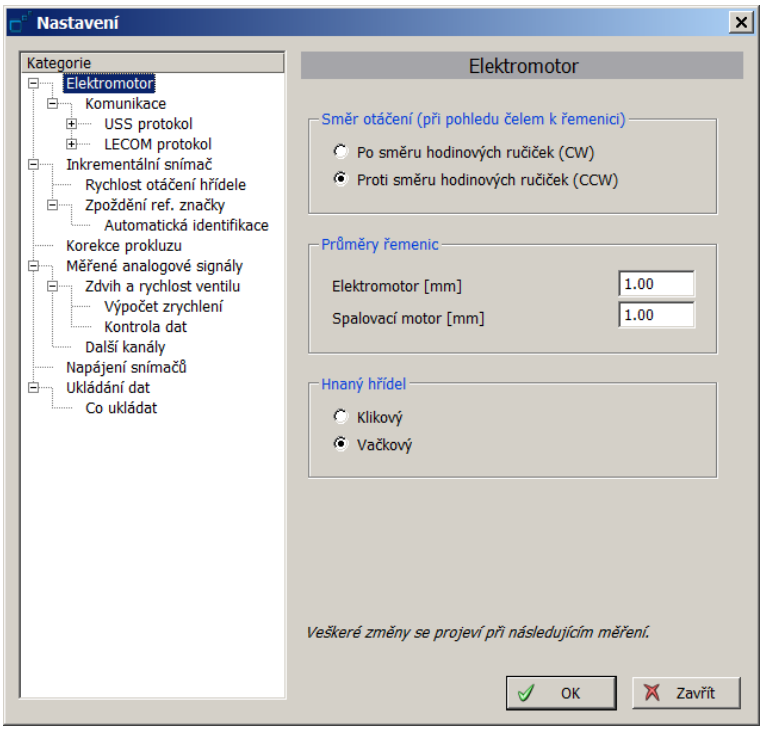


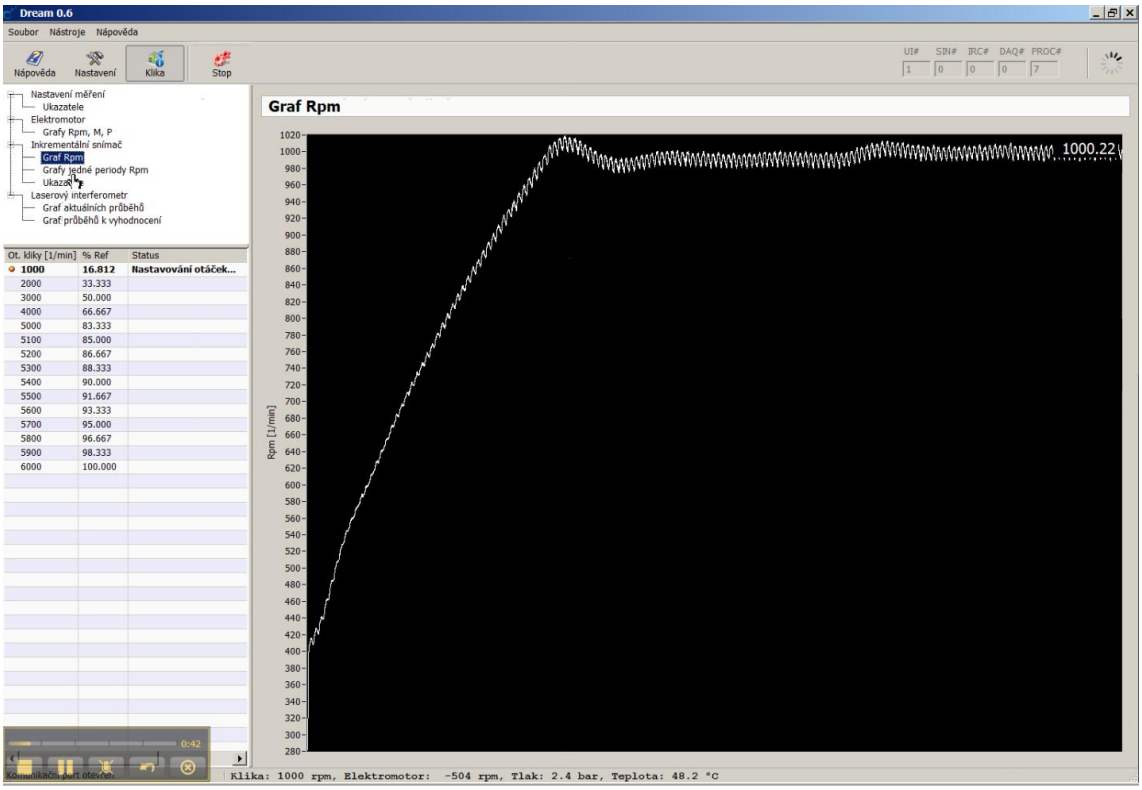
Dream printscreens:

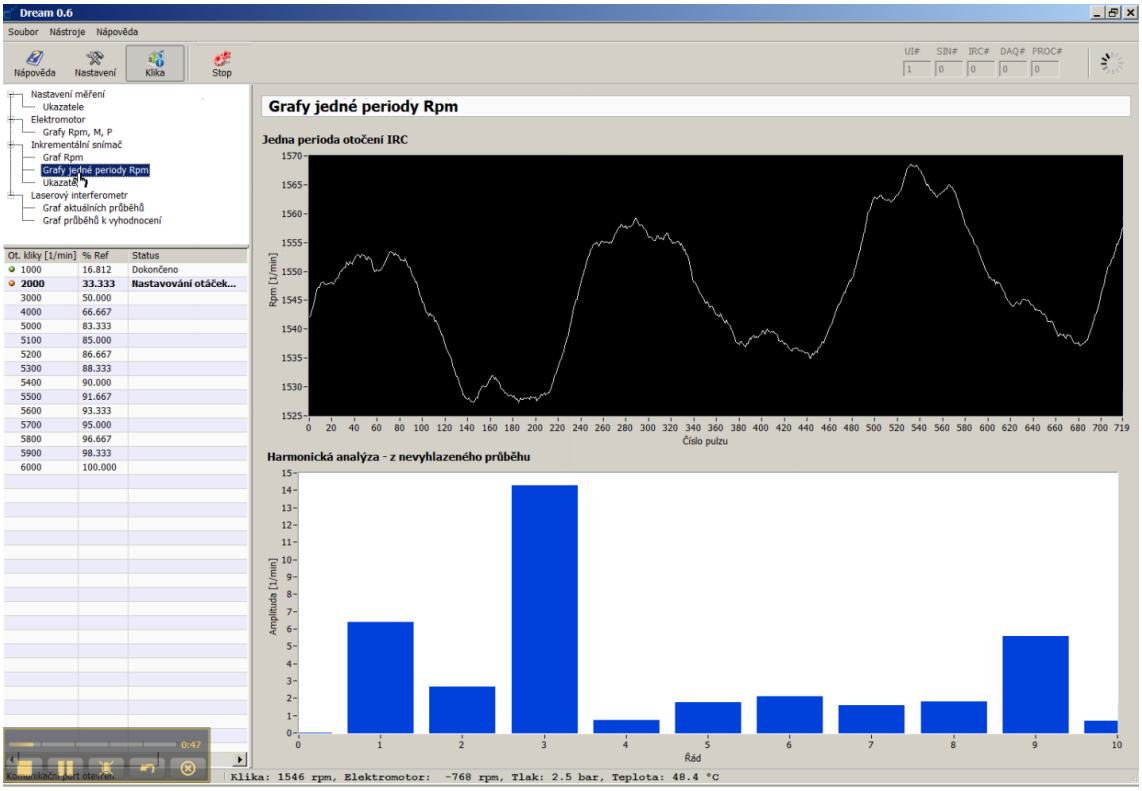


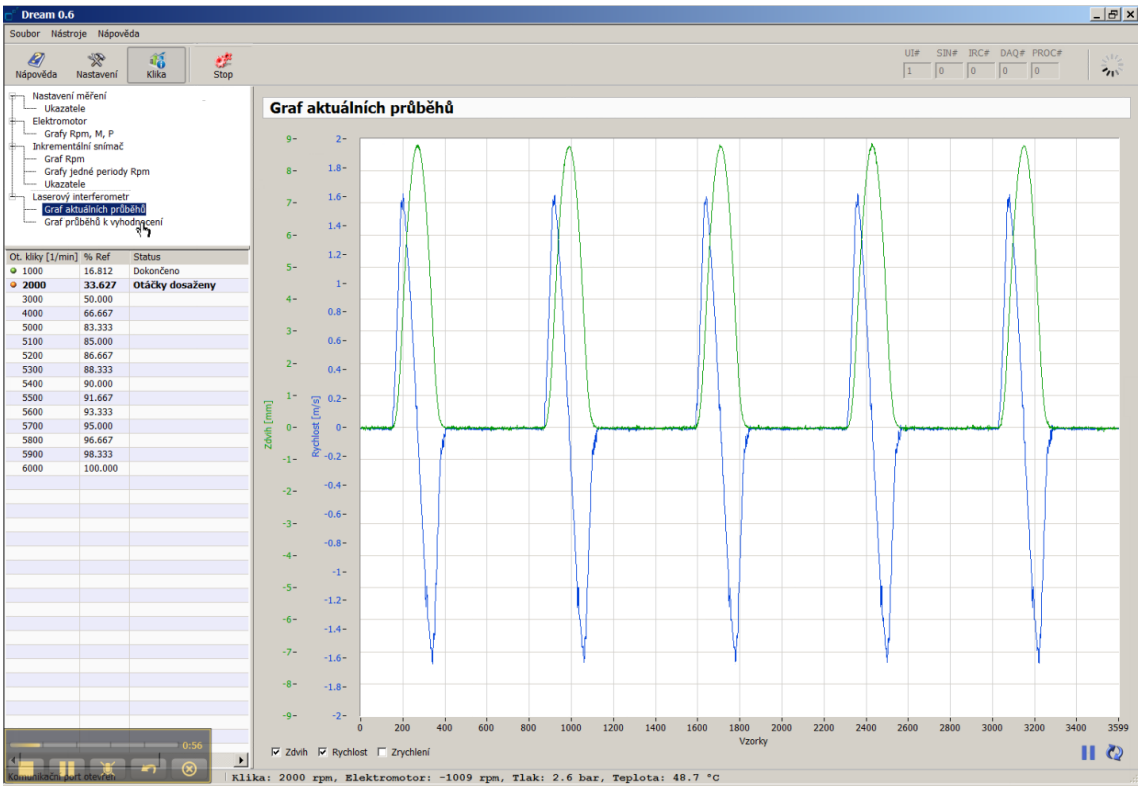


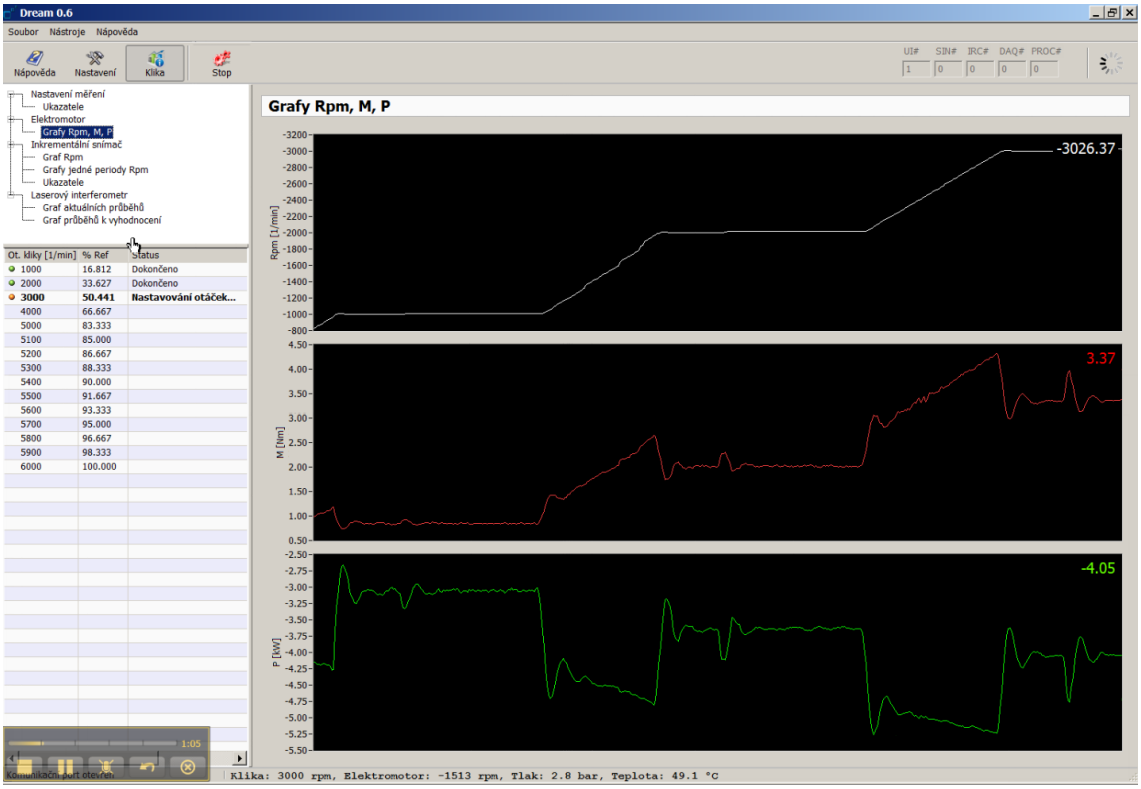

Dream 1.0 beta 3
Soubor Nástroje Nápověda
Nápověda Nastavení Originál Start
UI# ELM# IRC# DAQ# PROC#
ELM
ENG
Nastavení měření
Specifikace rozvodu
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Měřené veličiny
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Graf aktuálních průběhů
Ot. kliky [1/min] % Ref Status
1000 17.300
2000 33.333
3000 50.000
4000 66.667
5000 83.333
Zdvih [mm]
Rychlost [m/s]
Vzorky
Zdvih Rychlost Zrychlení kanál 3 kanál 4

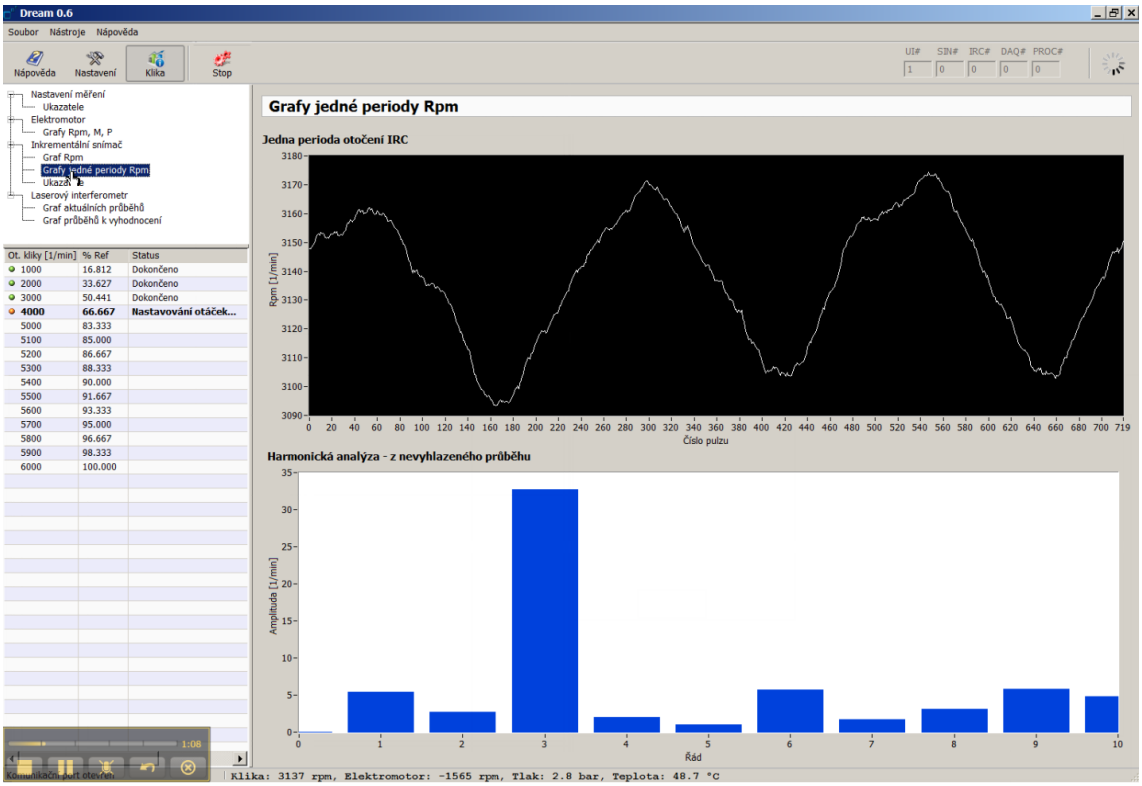

Nastavení
Kategorie
Elektromotor
Komunikace
USS protokol
LECOM protokol
Inkrementální snímač
Rychlost otáčení hřídele
Zpoždění ref. značky
Automatická identifikace
Korekce prokluzu
Měřené analogové signály
Zdvih a rychlost ventilu
Výpočet zrychlení
Kontrola dat
Další kanály
Napájení snímačů
Ukládání dat
Co ukládat
Elektromotor
Směr otáčení (při pohledu čelem k řemenici)
Po směru hodinových ručiček (CW)
Proti směru hodinových ručiček (CCW)
Průměry řemenic
Elektromotor [mm] 1.00
Spalovací motor [mm] 1.00
Hnaný hřídel
Klikový
Vačkový
Veškeré změny se projeví při následujícím měření.
OK
Zavřít

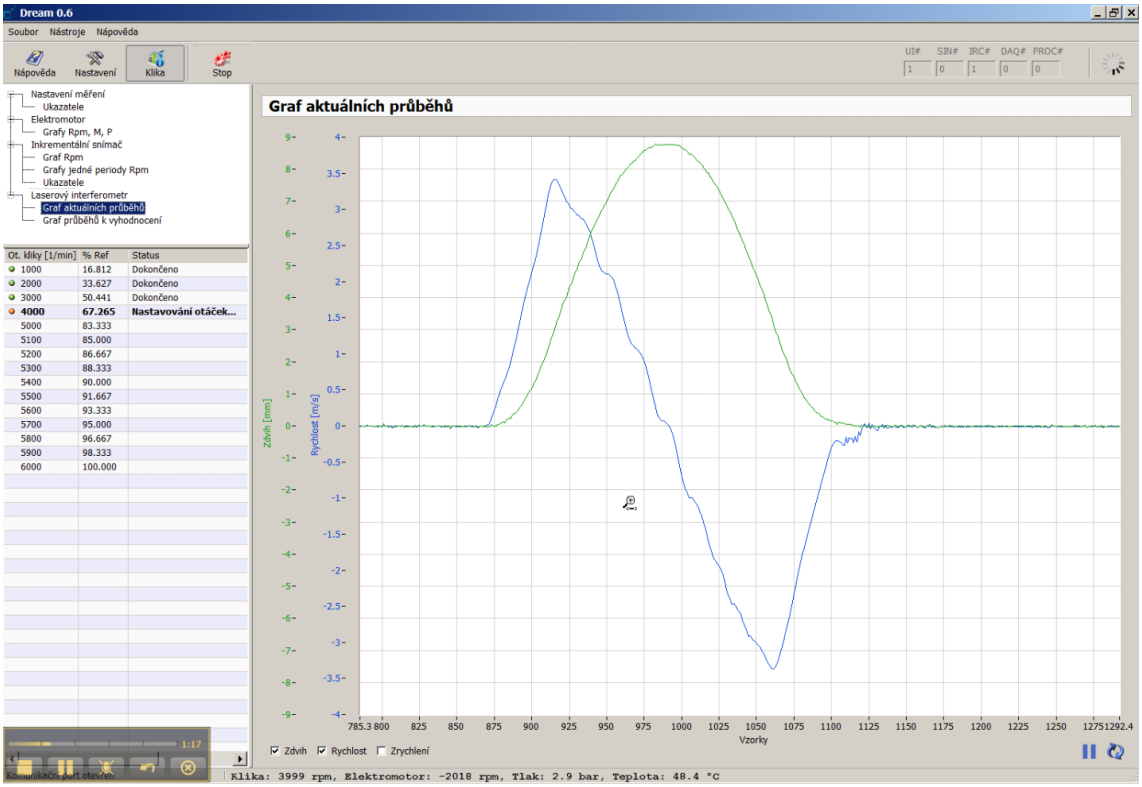

Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
UI# SIN# IRC# DAQ# PROC#
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Nastavování otáček...
2000 33.333
3000 50.000
4000 66.667
5000 83.333
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Graf Rpm
1000.22
Rpm [1/min]
Klika: 1000 rpm, Elektromotor: -504 rpm, Tlak: 2.4 bar, Teplota: 48.2 °C

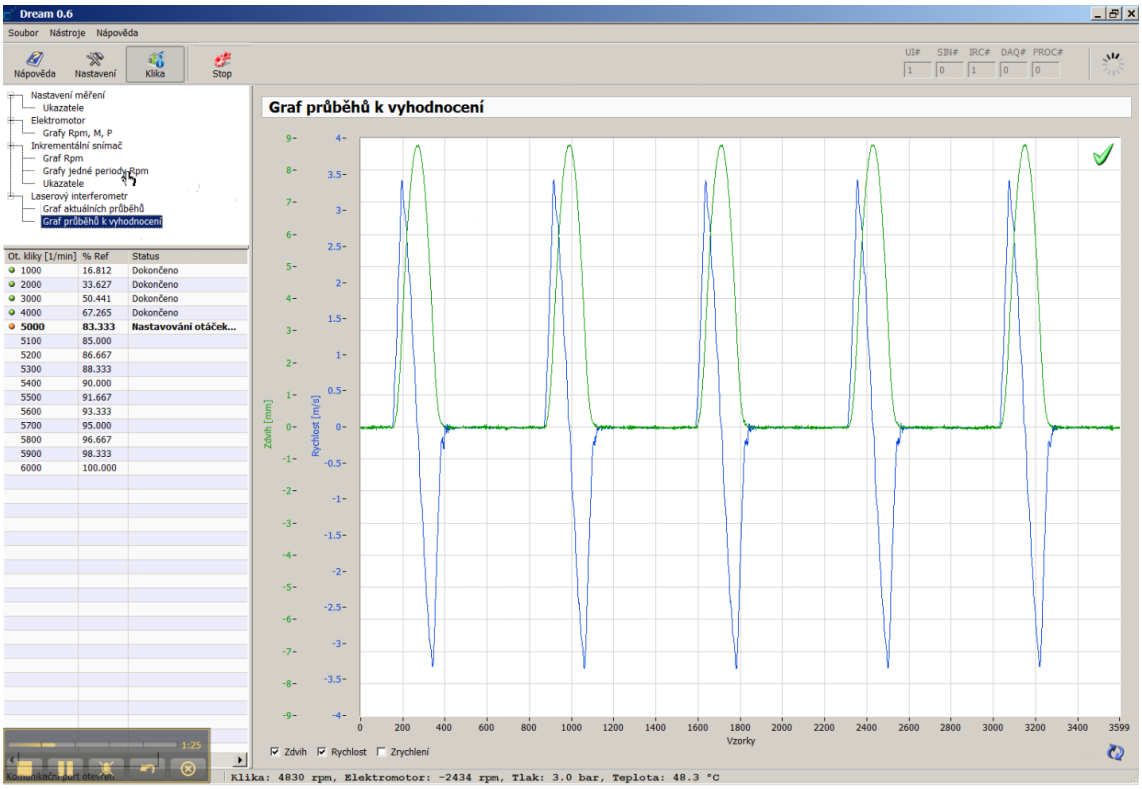

Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
Grafy jedné periody Rpm
Jedna perioda otočení IRC
Rpm [1/min]
Číslo pulzu
Harmonická analýza - z nevyhlazeného průběhu
Amplituda [1/min]
Řád
Klika: 1546 rpm, Elektromotor: -768 rpm, Tlak: 2.5 bar, Teplota: 48.4 °C

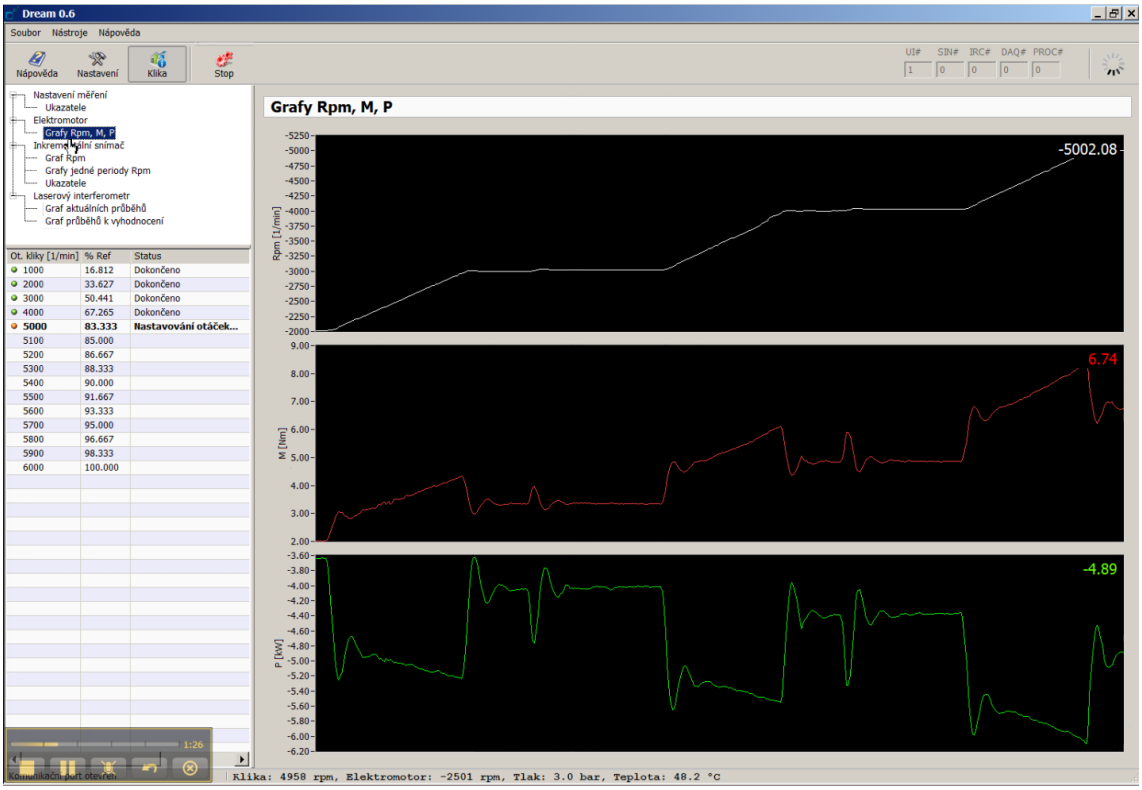

Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
UI# SIN# IRC# DAQ# PROC#
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Otáčky dosaženy
3000 50.000
4000 66.667
5000 83.333
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Graf aktuálních průběhů
Zdvih [mm]
Rychlost [m/s]
Vzorky
Zdvih Rychlost Zrychlení
Klika: 2000 rpm, Elektromotor: -1009 rpm, Tlak: 2.6 bar, Teplota: 48.7 °C


Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
UI# SIN# IRC# DAQ# PROC#
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Dokončeno
3000 50.441 Nastavování otáček...
4000 66.667
5000 83.333
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Grafy Rpm, M, P
Rpm [1/min]
-3026.37
M [Nm]
3.37
P [kW]
-4.05
Klika: 3000 rpm, Elektromotor: -1513 rpm, Tlak: 2.8 bar, Teplota: 49.1 °C


Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Dokončeno
3000 50.441 Dokončeno
4000 66.667 Nastavování otáček...
5000 83.333
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Grafy jedné periody Rpm
Jedna perioda otočení IRC
Rpm [1/min]
Číslo pulzu
Harmonická analýza - z nevyhlazeného průběhu
Amplituda [1/min]
Řád
Klika: 3137 rpm, Elektromotor: -1565 rpm, Tlak: 2.8 bar, Teplota: 48.7 °C


Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Dokončeno
3000 50.441 Dokončeno
4000 67.265 Nastavování otáček...
5000 83.333
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Graf aktuálních průběhů
Zdvih [mm]
Rychlost [m/s]
Vzorky
Zdvih
Rychlost
Zrychlení
Klika: 3999 rpm, Elektromotor: -2018 rpm, Tlak: 2.9 bar, Teplota: 48.4 °C


Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
UI# SIN# IRC# DAQ# PROC#
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Graf průběhů k vyhodnocení
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Dokončeno
3000 50.441 Dokončeno
4000 67.265 Dokončeno
5000 83.333 Nastavování otáček...
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Zdvih [mm]
Rychlost [m/s]
Vzorky
Zdvih Rychlost Zrychlení
Klika: 4830 rpm, Elektromotor: -2434 rpm, Tlak: 3.0 bar, Teplota: 48.3 °C


Dream 0.6
Soubor Nástroje Nápověda
Nápověda Nastavení Klika Stop
UI# SIN# IRC# DAQ# PROC#
Nastavení měření
Ukazatele
Elektromotor
Grafy Rpm, M, P
Inkrementální snímač
Graf Rpm
Grafy jedné periody Rpm
Ukazatele
Laserový interferometr
Graf aktuálních průběhů
Graf průběhů k vyhodnocení
Grafy Rpm, M, P
Ot. kliky [1/min] % Ref Status
1000 16.812 Dokončeno
2000 33.627 Dokončeno
3000 50.441 Dokončeno
4000 67.265 Dokončeno
5000 83.333 Nastavování otáček...
5100 85.000
5200 86.667
5300 88.333
5400 90.000
5500 91.667
5600 93.333
5700 95.000
5800 96.667
5900 98.333
6000 100.000
Rpm [1/min]
-5002.08
M [Nm]
6.74
P [kW]
-4.89
Klika: 4958 rpm, Elektromotor: -2501 rpm, Tlak: 3.0 bar, Teplota: 48.2 °C

Appendix 4

Parameters of the drop-out recognition algorithm

$p_1=3$
$p_2=3$
$p_3=4$ *(for ppr = 720)*
$p_4=4$
$p_5=0.02$
$p_6=85$
$p_7=4.5$
$p_8=0.01$
$p_9=0.5$